



**Nuno José  
Oliveira e Sá**

**Comunicação em Sistemas de Iluminação Pública  
Inteligentes**





**Nuno José  
Oliveira e Sá**

## **Comunicação em Sistemas de Iluminação Pública Inteligentes**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Doutor Luís Filipe Mesquita Nero Alves, Professor auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e do Doutor José Luís Vieira Cura, Professor auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.



## **Dedicatória**

Dedico este trabalho a todos aqueles que sempre me apoiaram no meu caminho acadêmico até atingir este objetivo. Pelo constante apoio e motivação, um muito obrigado a todos.



## **O júri**

Presidente

**Professor Doutor Dinis Gomes de Magalhães dos Santos**

Professor Catedrático do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

**Professor Doutor Nuno José de Abreu e Sousa Cabete Gil**

Professor Adjunto do Departamento de Engenharia Eletrotécnica da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria

**Professor Doutor Luís Filipe Mesquita Nero Moreira Alves**

Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

**Professor Doutor José Luís Vieira Cura**

Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro





## **Agradecimentos**

Em jeito de agradecimento, gostaria de deixar aqui umas palavras a todos aqueles que me ajudaram e apoiaram a atingir esta importante meta na minha vida.

Em primeiro lugar, gostaria de agradecer à Universidade de Aveiro, ao Departamento de Electrónica, Telecomunicações e Informática, e ao Instituto de Telecomunicações pela cedência de espaços e de todos os materiais necessários para efetuar um bom estudo e um bom trabalho. Seguidamente, gostaria de agradecer ao Orientador Professor Luís Filipe Mesquita Nero Moreira Alves e ao Coorientador Professor José Luís Vieira Cura pelo lançamento de uma interessante proposta de dissertação com elevado nível de crescimento.

Ao Professor Luís Filipe Mesquita Nero Moreira Alves queria deixar um especial agradecimento por toda a disponibilidade apresentada no auxílio a esclarecimento de dúvidas e escrita do trabalho. Ao Professor José Luís Vieira Cura também gostaria de agradecer pela preciosa ajuda na revisão da tese de mestrado.

Gostaria também de agradecer a todas as pessoas que passaram pelo laboratório de circuitos integrados, mostrando-se sempre disponíveis para ajudar. Queria deixar um agradecimento muito especial ao Mestre Nuno Rafael Mendonça Lourenço pelo seu constante apoio e disponibilidade na ajuda nos mais variados problemas, das mais diversas áreas. Ao mestre Daniel Ribau Lourenço também gostaria de deixar um muito obrigado. Também a todos os responsáveis pelo apoio técnico no Instituto de Telecomunicações gostava de deixar um agradecimento especial por estarem sempre disponíveis a ajudar.

Por último queria deixar um muito obrigado à minha família. Aos meus pais que me apoiaram sempre no meu estudo e me proporcionaram sempre as melhores condições para o meu estudo. Ao meu irmão que também sempre me apoiou no meu caminho escolar, principalmente na minha passagem pela universidade de Aveiro.

Deixo a todos estas palavras sinceras em jeito de agradecimento, pela ajuda recebida.

Muito Obrigado!



## Palavras-chave

LED (Light-Emitting Diode) , Sensores, Comunicação, Iluminação pública inteligente, Eficiência, Controlo.

## Resumo

Esta dissertação faz uma abordagem a sistemas de iluminação pública inteligentes. O seu objetivo passa pela eficiência energética com o intuito de reduzir os gastos energéticos atribuídos a este tipo de iluminação. Os recentes desenvolvimentos nos LEDs de alta potência permitiram implementar novas soluções para a iluminação pública, e ao mesmo tempo aumentaram significativamente a eficiência energética desta. Os LEDs apresentam inúmeras vantagens em relação às tecnologias antigas, como: elevada eficiência; possibilidade de variação do fluxo luminoso; tempo de vida elevado e mais ecológicos.

O presente estudo pretende fazer a continuação de um estudo prévio, donde resultou um protótipo de iluminação baseado em LEDs, com a integração de sensores e controlo. Este trabalho parte deste ponto e foca-se no conceito de sistemas de iluminação pública inteligentes. Assim, foi feito um estudo do protótipo desenvolvido e de todas as suas características, de topologias e opções de controlo e um estudo sensorial. No entanto, o grande foco deste trabalho passa pelo estudo de sistemas de comunicação com vista à integração destes em sistemas de iluminação. Para o efeito, foi feito um estudo de topologias e arquiteturas de comunicação, e um estudo de diferentes meios para a transmissão de dados. Partindo do protótipo de referência é objetivo desta tese de mestrado, alargar o seu conjunto de sensores e replicar o protótipo com vista à integração de comunicação entre duas luminárias e um terminal. Foi também possível desenhar, construir e validar experimentalmente um protótipo final em placas de circuito impresso (PCB). Assim, todo este trabalho estuda e valida o conceito de iluminação pública inteligente com recurso à tecnologia LED, tendo em especial consideração a integração de comunicação em sistemas deste tipo, possibilitando maior eficiência.



**Keywords**

LED (Light-Emitting Diode), Sensors, Communication, Intelligent Street Lighting, Efficiency, Control.

**Abstract**

This dissertation makes an approach to intelligent lighting systems. Its purpose passes through energy efficiency with the objective to reduce the energy that these systems spend.

The recent development of high power LEDs has allowed the study of new solutions for street lighting, having the potential of significantly increase the energy efficiency. There are many advantages in use light sources with high power LEDs, relatively to the oldest technologies, such as: high efficiency; possibility of luminous flux variation; high lifetime and more ecological. The present study has a previous study as starting point. As the result of the previous study was developed a prototype of lighting based on LEDs with the integration of sensors and control. At this point, the present work intends to study the intelligent lighting systems. Thus, was made a study of the developed prototype and all its features, of topologies and control options and a sensorial study. However, the major focus of this work passes through the study of communications systems. For this purpose, was made a study about topologies and architectures of communications, and mediums where the data can be transmitted. Having the reference prototype as starting point, is objective of this master's thesis, increase its sensorial features and replicate the prototype with the purpose of integrate communication between the two luminaries and the terminal. Besides all the theoretical study and all the design it was possible under practical work to develop printed circuit boards (PCB) used in the final prototype. Thus, all this work aims to learn and validate the concept of intelligent lighting systems based on LED technology, having special consideration about the implementation of communication in systems of this kind, enabling greater efficiencies.



# ÍNDICE

|   |     |
|---|-----|
| ÍNDICE .....  | i   |
| ÍNDICE DE FIGURAS .....                               | iii |
| ÍNDICE DE TABELAS .....                               | vii |
| LISTA DE ACRÓNIMOS .....                              | ix  |
| Capítulo I .....                                      | 1   |
| 1 Introdução .....                                    | 1   |
| 1.1 Enquadramento .....                               | 2   |
| 1.2 Motivações .....                                  | 2   |
| 1.3 Objetivos .....                                   | 5   |
| 1.4 Estrutura .....                                   | 5   |
| Capítulo II .....                                     | 7   |
| 2 Iluminação pública – Estado da arte .....           | 7   |
| 2.1 História dos sistemas de iluminação .....         | 8   |
| 2.2 Tecnologias de iluminação .....                   | 9   |
| 2.2.1 Lâmpadas de vapor de sódio .....                | 9   |
| 2.2.2 Lâmpadas de iodetos metálicos .....             | 10  |
| 2.2.3 Díodo emissor de luz (LED) .....                | 10  |
| 2.2.4 Comparativo entre tecnologias .....             | 12  |
| 2.3 Sistemas de Iluminação pública inteligentes ..... | 14  |
| 2.4 Casos de estudo .....                             | 16  |
| 2.5 Sistemas de controlo .....                        | 17  |
| 2.5.1 Unidade de controlo .....                       | 19  |
| 2.5.2 Arquiteturas de controlo .....                  | 19  |
| 2.5.3 Opções de controlo .....                        | 21  |
| 2.5.4 Topologias de controlo .....                    | 21  |
| 2.5.5 Gestão remota .....                             | 24  |
| 2.6 Sensores e variáveis de medida .....              | 25  |
| 2.6.1 Sensores Operativos .....                       | 26  |
| 2.6.2 Sensores de Manutenção .....                    | 27  |
| 2.6.3 Sensores Climatéricos .....                     | 28  |

|              |  |     |
|--------------|--|-----|
| 2.6.4        | Sensores de Integridade do sistema ..... | 29  |
| 2.7          | Integração de comunicação .....          | 30  |
| 2.8          | Em resumo.....                           | 32  |
| Capítulo III | .....                                    | 33  |
| 3            | Redes de comunicação .....               | 33  |
| 3.1          | Topologias de rede .....                 | 34  |
| 3.2          | Arquiteturas de rede.....                | 37  |
| 3.3          | Meio Físico.....                         | 39  |
| 3.3.1        | Wireless .....                           | 39  |
| 3.3.2        | <i>Power Line Communications</i> .....   | 40  |
| 3.3.3        | RS-485.....                              | 42  |
| 3.3.4        | RS-232.....                              | 45  |
| 3.4          | Interface de comunicação.....            | 46  |
| 3.4.1        | USART .....                              | 46  |
| 3.5          | Em resumo.....                           | 50  |
| Capítulo IV  | .....                                    | 53  |
| 4            | Sistema final.....                       | 53  |
| 4.1          | Protótipo de referência .....            | 54  |
| 4.2          | Detetor de falhas nos LEDs.....          | 56  |
| 4.3          | Sensor de <i>Tilt</i> .....              | 61  |
| 4.4          | Integração de Comunicação.....           | 69  |
| 4.4.1        | Hardware implementado .....              | 69  |
| 4.4.2        | <i>Software</i> .....                    | 73  |
| 4.4.3        | Implementação.....                       | 81  |
| 4.4.4        | <i>Master</i> .....                      | 85  |
| 4.4.5        | <i>Slave</i> .....                       | 91  |
| 4.5          | Em resumo.....                           | 94  |
| Capítulo V   | .....                                    | 97  |
| 5            | Conclusões.....                          | 97  |
| 5.1          | Análise de sensores.....                 | 97  |
| 5.2          | Análise da comunicação .....             | 99  |
| 5.3          | Trabalho futuro.....                     | 100 |
| Referências  | .....                                    | 101 |
| Anexos       | .....                                    | 103 |



# ÍNDICE DE FIGURAS

|   |    |
|---|----|
| Figura 1 – Exemplo de uma estrada iluminada com LEDs [1].                                   | 3  |
| Figura 2 - Comparativo entre gastos energéticos [1].  | 4  |
| Figura 3 – Análise espectral de um LED branco [11].   | 11 |
| Figura 4 – Constituição do <i>Golden Dragon Oval Plus</i> [13].                             | 12 |
| Figura 5 – Diagrama genérico de um sistema de iluminação inteligente.                       | 15 |
| Figura 6 – Esquema dos pinos do PIC18F258.  | 19 |
| Figura 7 - Controlo autónomo.   | 20 |
| Figura 8 - Controlo centralizado.   | 20 |
| Figura 9 - Gestão remota.   | 21 |
| Figura 10 – Topologia de um sistema autónomo com controlo passivo de energia.               | 22 |
| Figura 11 – Topologia de um sistema autónomo com controlo ativo de energia.                 | 22 |
| Figura 12 – Topologia de um sistema centralizado com controlo passivo de energia.           | 23 |
| Figura 13 – Topologia de um sistema centralizado com controlo ativo de energia.             | 23 |
| Figura 14 – Exemplo de um sistema utilizando gestão remota.                                 | 24 |
| Figura 15 – Ilustração da utilização de um sensor para controlar um conjunto de luminárias. | 25 |
| Figura 16 – Classificação dos sensores.   | 26 |
| Figura 17 – Diagrama de blocos de um sistema com comunicação.                               | 31 |
| Figura 18 – Topologia de rede em barramento.  | 34 |
| Figura 19 – Topologia ponto-a-ponto.  | 34 |
| Figura 20 – Topologia de rede em estrela.   | 35 |
| Figura 21 – Topologia de rede em anel.  | 36 |
| Figura 22 – Topologia de rede em árvore.  | 36 |
| Figura 23 – Exemplo de uma rede em malha e da sua robustez [17].                            | 37 |
| Figura 24 – Arquitetura <i>master-slave</i> .   | 37 |
| Figura 25 – Arquitetura <i>peer-to-peer</i> .   | 38 |
| Figura 26 – Arquitetura centralizada.   | 38 |
| Figura 27 – Exemplo de uma rede ZigBee [17].  | 40 |
| Figura 28 – Evolução das tecnologias PLC [25].  | 41 |
| Figura 29 – Esquemático de um <i>transceiver</i> RS-485 [28].                               | 43 |
| Figura 30 – Diagrama de blocos de um sistema com RS-485.                                    | 44 |
| Figura 31 – Exemplo de uma comunicação via RS-232.  | 46 |

|   |    |
|---|----|
| Figura 32 – Registo TXSTA [18]. .....   | 47 |
| Figura 33 – Registo RXSTA [18]. .....   | 47 |
| Figura 34 – Diagrama de blocos relativo ao transmissor [18]. .....                                | 49 |
| Figura 35 - Diagrama de blocos relativo ao recetor [18]. .....                                    | 49 |
| Figura 36 – Arquitetura implementada [32]. .....  | 54 |
| Figura 37 – Característica I-V dos TRIACS. ....   | 56 |
| Figura 38 – Diagrama correspondente à implementação do detetor de falhas.....                     | 58 |
| Figura 39 – Circuito básico de conversão de gamas. ....   | 59 |
| Figura 40 – Calibração do detetor de falhas. ....   | 60 |
| Figura 41 - Sensor de <i>tilt</i> utilizado [36]. ....  | 62 |
| Figura 42 – Funcionamento do sensor de <i>tilt</i> . ....   | 62 |
| Figura 43 – Configuração utilizando dois sensores. ....   | 63 |
| Figura 44 – Configuração utilizando 3 sensores. ....  | 64 |
| Figura 45 – Esquemático de como foram obtidos os resultados. ....                                 | 65 |
| Figura 46 – Gráfico com os resultados obtidos. ....   | 66 |
| Figura 47 – Esquemático da interligação com a unidade de controlo. ....                           | 67 |
| Figura 48 – Fluxograma do software do sensor de inclinação. ....                                  | 69 |
| Figura 49 - Diagrama de blocos referente ao master do sistema. ....                               | 70 |
| Figura 50 - Esquemático do circuito de seleção [37]. ....   | 71 |
| Figura 51 – Arquitetura do sistema final. ....  | 72 |
| Figura 52 - Localização do protocolo Modbus de acordo com o modelo OSI [30]. ....                 | 73 |
| Figura 53 - Diagrama correspondente a uma comunicação Modbus. ....                                | 74 |
| Figura 54 – <i>Application data unit</i> . ....   | 74 |
| Figura 55 – <i>Protocol Data Unit</i> . ....  | 75 |
| Figura 56 – Exemplo de um pedido de uma leitura de <i>coils</i> . ....                            | 76 |
| Figura 57 – Exemplo de uma resposta de uma leitura de <i>coils</i> . ....                         | 77 |
| Figura 58 – Exemplo de um pedido de uma escrita de uma <i>coil</i> . ....                         | 77 |
| Figura 59 – Exemplo de um pedido de uma leitura de um <i>holding register</i> . ....              | 78 |
| Figura 60 – Exemplo de uma resposta de uma leitura de um <i>Holding Register</i> . ....           | 78 |
| Figura 61 – Exemplo de um request de uma escrita de um único registo. ....                        | 79 |
| Figura 62 – Fluxograma que exemplifica o processamento de um request pelo slave [38]. ...         | 80 |
| Figura 63 – Mensagem de resposta do <i>slave</i> em caso de excepção. ....                        | 81 |
| Figura 64 – Fluxograma correspondente ao início do protocolo até à preparação do pedido.<br>..... | 86 |
| Figura 65 – Fluxograma relativo à transmissão de uma trama. ....                                  | 88 |

|   |           |
|---|-----------|
| <b>Figura 66 – Fluxograma relativo à rotina de atendimento à interrupção do <i>timer0</i>. .....</b>  | <b>90</b> |
| <b>Figura 67 - Fluxograma relativo à recepção de uma trama. ....</b>                                  | <b>91</b> |
| <b>Figura 68 – Fluxograma relativo ao processamento de um pedido de leitura de <i>coils</i>. ....</b> | <b>94</b> |



# ÍNDICE DE TABELAS

|   |    |
|---|----|
| Tabela 1 – Vantagens e desvantagens da tecnologia LED .....   | 13 |
| Tabela 2 - Vantagens e desvantagens das lâmpadas de vapor de sódio .....                                  | 13 |
| Tabela 3 - Vantagens e desvantagens das lâmpadas de iodetos metálicos .....                               | 14 |
| Tabela 4 - Vantagens e desvantagens de sistemas inteligentes de iluminação pública com LEDs .....         | 16 |
| Tabela 5 [17] - Redução no consumo de energia com o uso de controlo inteligente .....                     | 18 |
| Tabela 6 [17] - Potência consumida e comparação de custos .....   | 18 |
| Tabela 7 – Vantagens e desvantagens da integração de comunicação .....                                    | 31 |
| Tabela 8 [25-26] – Vantagens e desvantagens da tecnologia PLC .....                                       | 42 |
| Tabela 9 – Transmissão [29] .....   | 44 |
| Tabela 10 – Receção [29] .....  | 44 |
| Tabela 11 – Descrição dos bits que constituem o registo TXSTA [18] .....                                  | 47 |
| Tabela 12 – Descrição dos bits que constituem o registo RXSTA [18]. .....                                 | 48 |
| Tabela 13 – Níveis obtidos para o detetor .....   | 61 |
| Tabela 14 – Resultados obtidos para o sensor de inclinação .....  | 66 |
| Tabela 15 – Principais funções em Modbus [38] .....   | 76 |
| Tabela 16 – Lista de exceções do sistema .....  | 81 |
| Tabela 17 – Associação das variáveis Modbus com as variáveis do sistema ( <i>coils</i> ) .....            | 83 |
| Tabela 18 – Associação das variáveis Modbus com as variáveis do sistema ( <i>Holding Register</i> ) ..... | 83 |



# Lista de acrónimos

## A

---

**ADC** – Analog – to - Digital Converter

**AC** – Alternating current (corrente alternada)

**ADU** – Application Data Unit

**ASCII** – American Code for Information Interchange

## B

---

**Boost** - Regulador de comutação do tipo step up (a tensão de saída é superior à de entrada)

**Bps** – Bits por segundo

## C

---

**CO<sub>2</sub>** – Carbon Dioxide

**CFL** - Compact Fluorescent Lamp (Energy Saving)

**CAN** – Controlled Área Network

**CRC** - Cyclical Redundancy Checking

## D

---

**DTE** - Data Terminal Equipment

**DCE** - Data Circuit-terminating Equipment

**DC** – Direct Current (corrente contínua)

## I

---

**IRC** – Índice de Reprodução Cromático

## **K**

---

**KWh** – Kilowatt Hora

## **L**

---

**LED** – Light Emitting Diode

**LITES** - Led-based Intelligent Street Lighting for  
Energy Saving

**LD** – Led Driver

**LVDT** – Linear Variable Differential Transformer

## **M**

---

**MSSP** – Master Synchronous Serial Port Module

## **O**

---

**OSI** – Open Systems Interconnection Model

## **P**

---

**PLC** – Power Line Communication

**PWM** - Pulse Width Modulation

**PN** - Junção semicondutora constituída por um material dopado  
positivamente e um material dopado negativamente

**PCB** – Printed Circuit Board

**PDU** – Protocol Data Unit

## **R**

---

**RGB** – Red Green Blue

**RTD** – Resistive Thermal Devices



**RVDT** - Rotary Variable Differential Transformer

**RS** - Recommended Standard

**RTU** – Remote Terminal Unit

## **S**

---

**SSL** – Solid State Lighting

**SI** - Sistema Internacional

## **V**

---

**VSAP** – Vapor de sódio de alta pressão

**VSBP** – Vapor de sódio de baixa pressão

## **U**

---

**USART** - Universal Synchronous Asynchronous Receiver  
Transmitter

**USB** – Universal Serial Port

## **T**

---

**TRIAC** - Triode for Alternating Current



# **Capítulo I**

## **1 Introdução**

A iluminação pública constitui um importante fator económico na sociedade atual, uma vez que é responsável por uma fatia considerável nos gastos energéticos mundiais.

Recentemente, tem-se vindo a observar o desenvolvimento de sistemas inteligentes de iluminação pública associado ao desenvolvimento da tecnologia LED (Light-Emitting-Diode). Estes sistemas, pretendem fazer uso da leitura de um conjunto de variáveis e da integração de comunicação para atuarem da forma mais eficiente. Apresentam vantagens como a monitorização à distância, regulação da temperatura dos LEDs e diminuição dos custos de manutenção.

O presente trabalho pretende dar continuidade a um outro previamente desenvolvido, onde já são introduzidos sensores para melhorar o funcionamento do sistema. Assim pretende-se alargar o número de sensores e fazer o estudo e integração de comunicação em sistemas de iluminação pública inteligentes com LEDs.

## **1.1 Enquadramento**

Este projeto de mestrado está integrado no projeto europeu LITES (*Led-based Intelligent Street Lighting for Energy Saving*), vocacionado para a eficiência energética a nível da iluminação pública. O projeto LITES tem como grande objetivo oferecer soluções inteligentes para iluminação pública recorrendo a dispositivos SSL (*Solid State Lighting*), nomeadamente o LED, com o intuito de reduzir drasticamente o consumo energético, cujo potencial de redução é estimado em cerca de 70% [1]. Neste pretende-se dar continuidade a um outro previamente desenvolvido, que tem como resultado final um protótipo, onde é implementado um sistema que já faz uso da tecnologia LED. Este trabalho já tem integrado sensores que permitem fazer uma melhor gestão dos gastos energéticos. O atual trabalho de mestrado pretende partir deste ponto, aumentando o leque de sensores disponíveis e possibilitar um aumento da “inteligência” do sistema, nomeadamente com a introdução de comunicação em sistemas de iluminação pública.

## **1.2 Motivações**

Atualmente, a eficiência energética é uma das grandes preocupações da sociedade moderna, por isso nos dias de hoje estamos a assistir à introdução de novas tecnologias, nas mais diversas áreas, que visam atingir esse objetivo. É neste panorama que se insere a temática da iluminação pública, uma vez que desde sempre constituiu uma das mais importantes formas de segurança em ambiente noturno e é responsável por cerca de 20% do consumo elétrico total. Sendo assim, torna-se necessário procurar opções que permitam minimizar os gastos na iluminação pública, visto que, nas últimas décadas, a necessidade de iluminação tem vindo a aumentar. É neste contexto que aparecem os dispositivos de estado sólido, nomeadamente os LEDs. Estes têm vindo a ser aperfeiçoados e, atualmente, já é possível encontrar LEDs de alta potência com níveis de fluxo luminoso bastante elevados e atingindo eficiências superiores à das tecnologias de iluminação mais comuns. De referir que presentemente, as lâmpadas de vapor de sódio ainda são as mais utilizadas em iluminação pública [2-3], apesar de a tecnologia LED estar a ganhar terreno. O aparecimento dos LEDs de alta potência que emitem luz branca marcam uma viragem na iluminação pública. Na figura 1 [1] observa-se o exemplo de uma estrada com iluminação a LED. É de realçar que a iluminação com luz branca é bastante mais uniforme do que é habitual com a iluminação com lâmpadas de vapor de sódio, em que, a luz amarelada não é

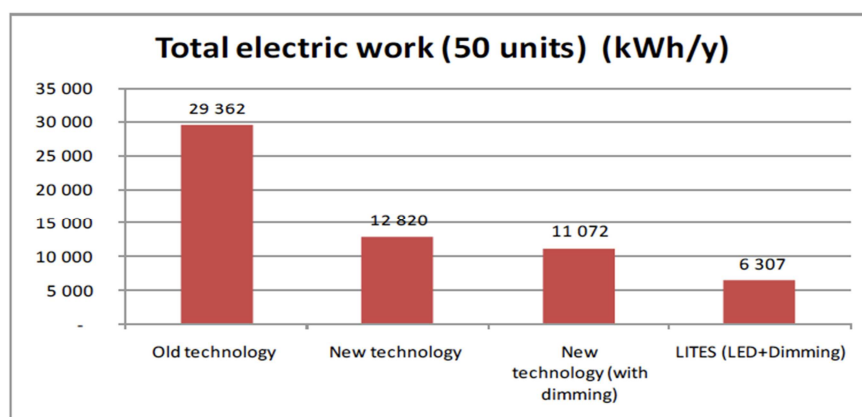
tão uniformemente distribuída. Assim, em comparação com outras tecnologias, os LEDs têm um maior tempo de vida e características luminosas iguais ou melhores, que proporcionam anos de operação livres de manutenção com um quantificável retorno no investimento [4].



**Figura 1 – Exemplo de uma estrada iluminada com LEDs [1].**

Em conjunto com a lâmpada LED, procuram-se também novas estratégias e opções que permitam fazer um uso mais inteligente da iluminação com vista a uma maior eficiência energética. Assim, os chamados sistemas de iluminação pública inteligentes, visam reduzir os custos energéticos fazendo uso das informações que o meio onde estão inseridos pode oferecer. Atualmente e, em maior parte dos casos, apenas se controla o ligar/desligar das luminárias consoante a luz ambiente que é detetada. No entanto, variáveis como a deteção de movimento são de extrema importância no âmbito da eficiência energética, principalmente em áreas mais rurais em que tráfego é bastante mais reduzido que em áreas urbanas. Neste contexto de sistemas inteligentes, a tecnologia LED apresenta grandes vantagens relativamente às outras tecnologias, uma vez que a integração de sensores em sistemas eletrónicos é relativamente fácil. Por outro lado, também possibilita facilmente a variação do fluxo luminoso. De referir que a lâmpada de vapor de sódio, atualmente, também permite a variação de fluxo luminoso através da introdução de balastros eletrónicos, no entanto, a sua integração em sistemas inteligentes é difícil, uma vez que apresentam tempos de arranque bastante elevados, contrariamente aos dos LEDs. Existem outras variáveis que podem ser utilizadas em sistemas deste género, como a temperatura dos LEDs. A temperatura dos LEDs é um importante fator a ter em conta, uma

vez que podem reduzir fortemente o tempo de vida destes. Assim torna-se necessário haver uma boa dissipação do calor. Adicionalmente, a introdução de sensores que possam fazer leituras da sua temperatura é de grande importância. A nível sensorial existe um leque alargado de opções que podem ser estudadas, implementadas e validadas em sistemas de iluminação pública inteligentes. Na figura seguinte [1] observa-se um comparativo, entre tecnologias, onde é evidente a vantagem, em termos de reduções energéticas, da tecnologia LED.



**Figura 2 - Comparativo entre gastos energéticos [1].**

Nesta temática de sistemas inteligentes, têm sido feitos alargados estudos com o propósito de lhes introduzir um sistema de comunicação. As vantagens são imensas pois, desde logo, se torna possível fazer a manutenção dos sistemas à distância, ou seja, é exequível fazer uma monitorização do sistema consoante os valores apresentados pelos seus sensores. Com a introdução de comunicação existe um grande número de opções de controlo que podem ser implementadas, surgindo de imediato o conceito de controlo remoto. Por outro lado existem sensores que poderiam ser comuns a um conjunto de luminárias, fazendo uso da comunicação, para partilharem os seus valores. Existem várias abordagens que podem ser feitas no âmbito da implementação de comunicação. Dentro desta temática, o projeto LITES pretende fazer uso da atual rede de distribuição elétrica para fazer a transferência de dados, ou seja fazer uso das PLC (*Power Line Communications*) [5].

Sistemas de iluminação pública inteligentes são sistemas que de facto podem introduzir uma mais-valia na atualidade energética mundial, sendo cada vez mais uma hipótese. Em Portugal e em outros países europeus já se tem vindo a observar a introdução

de iluminação recorrendo a LEDs. O principal obstáculo na inclusão deste tipo de sistemas é o investimento inicial.

### **1.3 Objetivos**

O objetivo desta tese de mestrado é fazer um estudo acerca de sistemas de iluminação pública inteligentes que fazem uso de dados sensoriais e de uma rede de comunicação para tornarem estes sistemas energeticamente mais eficientes e para facilitar a manutenção destes. Assim os objetivos desta dissertação são:

- ✓ Estudo bibliográfico acerca de LEDs de alta potência;
- ✓ Estudo de sensores, estratégias de controlo e essencialmente de redes de comunicação em sistemas deste tipo;
- ✓ Desenvolvimento e validação experimental de sensores úteis para sistemas de iluminação pública;
- ✓ Desenvolvimento e validação experimental de uma rede de comunicações para sistemas de iluminação pública;
- ✓ Integrar os dois pontos anteriores no desenvolvimento de um protótipo final.

### **1.4 Estrutura**

No presente capítulo pretende-se fazer uma introdução à temática abordada na dissertação, enquadramento da respetiva proposta e quais os objetivos que se pretendem atingir.

No segundo capítulo fala-se um pouco da história de iluminação, e também uma pequena abordagem às tecnologias de iluminação (dando mais ênfase ao LED) e faz-se a introdução aos sistemas de iluminação pública inteligente, fazendo uma abordagem a nível de controlo, sensorial e integração de comunicação.

No terceiro capítulo direciona-se o estudo para sistemas de comunicação. Aqui fala-se de quais as topologias de rede que se podem adotar, quais as arquiteturas possíveis e alguns meios em que os sinais se podem propagar. É também feito um estudo acerca de um interface de comunicação universal, a USART (Universal Synchronous Asynchronous Receiver Transmitter).

No quarto capítulo são apresentados todos os passos de forma a atingir o sistema final. Todas as escolhas feitas são discutidas neste capítulo, bem como todas as dificuldades e limitações encontradas. São expostas todas as implementações efetuadas no decurso do projeto e todos os resultados práticos obtidos no teste do sistema.

Finalmente, no quinto capítulo é feita a discussão de todos os resultados obtidos durante o teste do protótipo e quais as suas limitações. Conclui-se acerca da viabilidade do sistema e propõe-se a realização de trabalho futuro.



## **Capítulo II**

### **2 Iluminação pública – Estado da arte**

Desde há muito tempo que houve necessidade de manter as ruas iluminadas durante a noite, essencialmente por uma questão de segurança. Com o decorrer dos anos, as tecnologias de iluminação têm vindo a evoluir cada vez mais, sendo também mais eficazes. No entanto, ainda não são tão eficientes quanto o esperado e, com os recursos energéticos a escassear, torna-se cada vez mais importante fazer um uso cuidadoso da energia. Assim sendo, e uma vez que a iluminação pública é responsável por uma quantia considerável de gastos energéticos, torna-se de extrema importância atuar. É neste panorama que a tecnologia LED, pelas suas características, pode levar a uma redução substancial do consumo energético na iluminação pública.

Este capítulo será iniciado com uma breve história acerca da evolução dos sistemas de iluminação. Seguidamente serão apresentadas as tecnologias de iluminação que são mais utilizadas na atualidade e a tecnologia LED. Será também introduzido o conceito de sistemas de iluminação inteligentes e alguns casos de estudo da utilização de LEDs. Para finalizar falar-se-á de sistemas de controlo, integração de sensores e comunicação em sistemas de iluminação.

### **2.1 História dos sistemas de iluminação**

As primeiras lâmpadas a serem utilizadas, pelos gregos e romanos, foram lâmpadas de óleo, sendo utilizadas apenas com a finalidade de oferecer segurança. Eram utilizadas uma vez que eram duradouras e providenciavam uma chama moderada. No entanto, falando especificadamente de iluminação no âmbito da segurança na circulação de veículos, esta começou a aparecer na década de 1930 [6]. Três fatores convergiram para o aparecimento de iluminação nas ruas:

1. Aparecimento da tecnologia necessária, mais especificadamente, o aparecimento de uma rede de distribuição elétrica acompanhada com luminárias adequadas [6].
2. Estabelecimento oficial de sistemas de regulação de veículos e controlo de tráfego [6].
3. Crescimento do número de veículos e das suas velocidades [6].

Apesar da convergência destes três fatores, o crescimento da iluminação pública foi lento.

A discussão efetuada acima diz respeito a locais fora das localidades ou cidades pois a iluminação das estradas em áreas urbanas apareceu bastante mais cedo por uma questão de segurança dos habitantes. De facto, em Paris, no século XV, durante os meses de novembro, dezembro e janeiro era obrigatório colocar, antes das 6h da tarde, uma espécie de lanterna apontada para a estrada, ao nível da primeira janela, permanecendo esta durante toda a noite, numa posição adequada para poder fornecer luz suficiente para a estrada [6]. Com o crescimento das cidades, houve cada vez mais procura por algum tipo de iluminação, aparecendo então a iluminação a gás, que foi a maior fonte de iluminação à noite durante aproximadamente 100 anos. Em Londres, em 1823, a iluminação a gás tinha crescido tanto que cerca de 39,000 lâmpadas de gás forneciam luz para 215 milhas de estrada urbana [6]. Só no início do século XX é que as lâmpadas a gás começaram a desaparecer por completo, dando então lugar à iluminação elétrica. A primeira estrada eletricamente iluminada no Reino Unido foi a Mosley Street, em Newcastle, no ano de 1879, utilizando a lâmpada incandescente de Joseph Swan [7]. A utilização de lâmpadas incandescentes começou a crescer nos finais do século XIX, inícios do século XX e permaneceram na iluminação pública até ao aparecimento das lâmpadas de descarga gasosa. Atualmente as lâmpadas mais utilizadas na iluminação pública são as lâmpadas de

vapor de sódio de alta pressão [2-3]. No entanto, hoje em dia começam a emergir novas tecnologias, como a tecnologia LED, sobretudo após o aparecimento do LED branco de alta potência. Este é, de facto, um dispositivo que promete ser o futuro na iluminação pública e proporcionar uma revolução a nível de gastos energéticos. Na subsecção 2.2.3 será feito um estudo detalhado desta tecnologia. Muito recentemente introduziu-se o conceito de sistemas inteligentes de iluminação pública que tem como objetivo maximizar a eficiência na iluminação pública, reduzindo o consumo energético.

## **2.2 Tecnologias de iluminação**

Nesta secção irá ser feita uma breve abordagem a duas tecnologias de iluminação, que, atualmente, são as mais utilizadas na iluminação pública. Essas tecnologias são as lâmpadas de vapor de sódio e as lâmpadas de iodetos metálicos. Seguidamente irá ser feita uma abordagem bem mais detalhada acerca do LED, tecnologia que tem vindo a ganhar terreno nos últimos anos relativamente às anteriores. Para finalizar será feito um comparativo entre estas três tecnologias.

### **2.2.1 Lâmpadas de vapor de sódio**

As lâmpadas de vapor de sódio incluem-se na categoria de lâmpadas que se baseiam na descarga de um gás ou vapor metálico (a alta ou baixa pressão) para produzirem luz e são a tecnologia mais usada na atualidade [2-3], apesar de estarem a perder terreno para os LEDs.

As lâmpadas de vapor de sódio podem ser divididas em duas categorias com características distintas, sendo elas:

- ✓ Lâmpadas de vapor de sódio de baixa pressão.
- ✓ Lâmpadas de vapor de sódio de alta pressão (VSAP).

As lâmpadas de sódio de baixa pressão consistem num invólucro de vidro transparente, capaz de manter o vácuo interno, revestido por uma fina camada de material transparente para a luz visível mas refletor de infravermelhos. Este invólucro permite assim manter a atmosfera extremamente rarefeita, possibilitando a formação de plasma de vapor de sódio, permitindo a saída da luz visível e mantendo a radiação infravermelha no seu interior. Outra característica importante deste tipo de lâmpadas é o facto de estas apresentarem uma elevada eficiência energética, uma vez que a sua emissão se faz numa

banda bastante estreita, em torno de um comprimento de onda ao qual o olho humano é bastante sensível, sendo a luz percebida como amarela. No entanto apresentam a desvantagem de possuir um baixíssimo valor de IRC (Índice de Reprodução Cromática). IRC é um valor que pode variar de 0 a 100 conforme a radiação da fonte luminosa se distribui, menos ou mais, na gama do visível.

Relativamente às de alta pressão, as lâmpadas de vapor de sódio são menores e contêm elementos químicos adicionais, como o mercúrio. No entanto, estas possuem uma constituição idêntica em relação às anteriores, contudo as altas pressões existentes permitem alargar substancialmente a largura de banda da emissão, levando a um IRC mais elevado (da ordem dos 85) [8]. De notar que as lâmpadas de sódio de alta pressão, tal como as de baixa pressão, necessitam da presença de um gás ionizável para despoletar o arranque, sendo normalmente utilizado o xénon em muito baixa pressão. De referir que é possível a variação do fluxo luminoso se balastros eletrónicos forem integrados.

### **2.2.2 Lâmpadas de iodetos metálicos**

A par das lâmpadas de vapor de sódio, as lâmpadas de iodetos metálicos são das mais utilizadas em iluminação pública, sendo também bastante utilizadas nos faróis dos veículos [6]. Estas são uma fonte de luz baseadas nas lâmpadas de vapor de mercúrio, no entanto diferem destas uma vez que são usados iodetos metálicos, como iodetos de sódio e escândio. Nas temperaturas de operação, os iodetos metálicos são vaporizados, separando-se em metais e halogéneos, sendo os metais responsáveis pela emissão de radiação na zona visível do espectro.

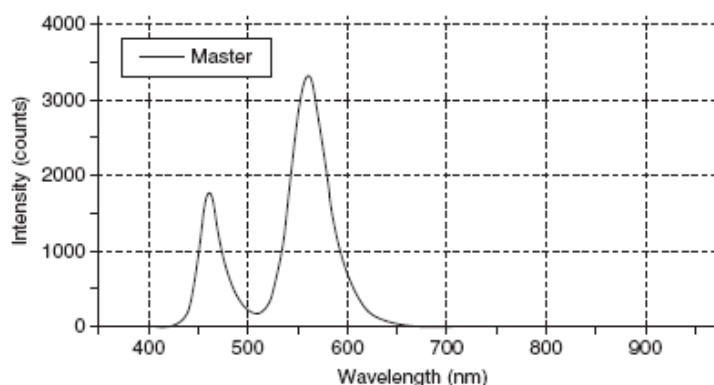
### **2.2.3 Díodo emissor de luz (LED)**

Aqui será feita uma descrição detalhada da tecnologia LED, apontada como o futuro não só na iluminação pública, mas como da iluminação em geral.

A tecnologia básica que está por detrás do aparecimento do LED apareceu nos anos 60, quando cientistas trabalhavam com um *chip* semicondutor, dopando-o com impurezas de modo a criar uma junção p-n [9]. Tal como nos díodos, a corrente no LED circula da região p (ânodo) para a região n (cátodo). À semelhança do díodo, no LED, facilmente se cria uma corrente que flui da região tipo p para a região tipo n, mas não no sentido inverso [9]. A criação de luz acontece quando um eletrão atravessa a barreira de potencial da zona

de depleção e encontra um buraco, caindo assim num nível de energia mais baixo, resultando na libertação de um fóton (libertando energia). Um fóton não é nada mais do que um portador de radiação eletromagnética de todos os comprimentos de onda. O comprimento de onda da luz que é gerada bem como a sua cor dependem da energia de *BandGap* dos materiais usados para formar a região p-n. Entenda-se como *BandGap* a energia que é necessária ultrapassar para que os eletrões se movam da banda de valência para a banda de condução [10].

O LED é então uma fonte de luz no estado sólido (*Solid State Light*) que já tem vindo a substituir as lâmpadas incandescentes em algumas áreas, como por exemplo em sinais de trânsito [6]. A luz produzida pelo LED é determinada pela corrente que o atravessa bem como pela temperatura a que este se encontra. O tempo de vida de um LED pode variar entre uns poucos milhares de horas até às 100.000 horas, dependendo da corrente utilizada e das condições em que estes operam [6]. Estes normalmente produzem radiação numa banda estreita, o que pode levar a pensar que o seu uso onde luz branca é requerida não é possível. No entanto não é bem assim, uma vez que é possível produzir luz branca recorrendo a duas técnicas, utilizando LEDs. Uma delas consiste em combinar a luz de saída de LEDs vermelhos, azuis e verdes (RGB) nas proporções corretas. A outra consiste em fazer passar a luz azul por fósforo. Uma vez conseguida a luz branca é então possível a utilização desta tecnologia no âmbito da iluminação pública. De referir que também é possível implementar um LED que emite diretamente luz branca, utilizando para isso seleneto de zinco (ZnSe), possuindo um pico aos 460 nm e outro aos 560 nm como é possível observar na figura 3 [11].



**Figura 3 – Análise espectral de um LED branco [11].**

Especial atenção deve ser dada à corrente de polarização dos LEDs, uma vez que existe uma relação direta entre esta e a temperatura do LED [12]. Sendo assim é necessário manter a temperatura da junção do LED baixa, de modo a obter uma maior eficiência e garantir maior tempo de vida do LED. De uma forma muito sucinta serão descritas duas formas de atingir este objetivo. Por um lado a luz de saída deve ser controlada em torno de um mínimo de luminância aceitável, de maneira a reduzir o consumo de potência e potenciar maior tempo de vida do componente. Por outro lado, pode ser feito um controlo pulsado (*Pulse Width Modulation*) da corrente que atravessa o LED, ou seja, variar o tempo em que o LED está efetivamente a conduzir.

Esta é uma tecnologia que está a emergir, sendo apontada como o futuro da iluminação pública. Na figura 4 [13] e a título de exemplo está demonstrada a constituição de um LED relativamente recente (*Golden Dragon Oval Plus*), possuindo neste caso uma lente para tornar a luz mais diretiva.

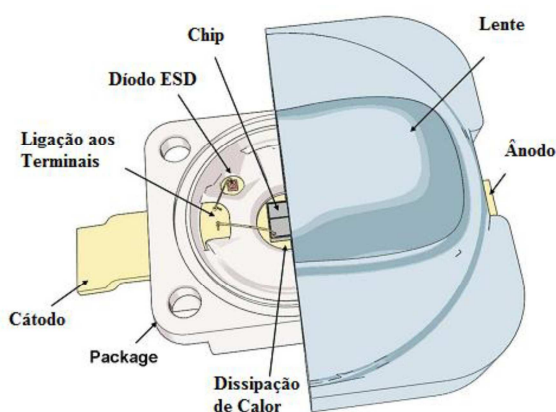


Figura 4 – Constituição do *Golden Dragon Oval Plus* [13].

### 2.2.4 Comparativo entre tecnologias

Nesta subsecção será feito um comparativo entre as três tecnologias apresentadas previamente. Nesse sentido nas tabelas 1, 2 e 3 é então apresentado um resumo das vantagens e desvantagens das três tecnologias descritas previamente.

Fazendo uma apreciação do conjunto das seguintes três tabelas é então possível concluir, que, as lâmpadas LED têm, de facto, potencial para se insurgir no mercado da iluminação pública.

| <u>Lâmpada LED</u>  |  |
|---|--|
| Vantagens   | Desvantagens   |
| <ul style="list-style-type: none"> <li>• IRC elevado;</li> <li>• Eficiência energética elevada;</li> <li>• Melhor qualidade da luz;</li> <li>• Facilidade na integração de sensores;</li> <li>• Tamanho reduzido;</li> <li>• Arranque imediato;</li> <li>• Facilidade na variação do fluxo luminoso;</li> <li>• Resistência a impactos e vibrações;</li> <li>• Tempo de vida elevado;</li> <li>• Diminuição da poluição;</li> <li>• Sem partes móveis, mercúrio, gases e nenhum filamento.</li> </ul> | <ul style="list-style-type: none"> <li>• Custo de aquisição relativamente elevado;</li> <li>• Necessidade de dispositivos de dissipação de calor;</li> <li>• Dependência da temperatura;</li> <li>• Alimentação DC.</li> </ul> |

Tabela 1 – Vantagens e desvantagens da tecnologia LED

| <u>Lâmpadas de vapor de sódio</u>   |  |
|---|--|
| Vantagens   | Desvantagens   |
| <ul style="list-style-type: none"> <li>• IRC aceitável (alta pressão);</li> <li>• Durabilidade elevada;</li> <li>• Variação de fluxo luminoso, com integração de balastros eletrônicos;</li> <li>• Elevada eficiência energética;</li> <li>• Capacidade de manter o fluxo luminoso durante toda a sua vida útil (baixa pressão);</li> </ul> | <ul style="list-style-type: none"> <li>• Arranque lento;</li> <li>• IRC reduzido (baixa pressão);</li> <li>• Distorção das cores pela emissão monocromática;</li> <li>• Equipamento auxiliar.</li> </ul> |

Tabela 2 - Vantagens e desvantagens das lâmpadas de vapor de sódio

| Lâmpadas de iodetos metálicos   |   |
|---|---|
| Vantagens   | Desvantagens  |
| <ul style="list-style-type: none"><li>• IRC bom;</li><li>• Baixa carga elétrica;</li><li>• Elevada eficiência energética;</li><li>• Temperatura de cor (3000K a 4900K);</li></ul> | <ul style="list-style-type: none"><li>• Caras;</li><li>• Equipamento auxiliar;</li><li>• Não admitem variações de tensão superiores a 5%.</li></ul> |

Tabela 3 - Vantagens e desvantagens das lâmpadas de iodetos metálicos

### 2.3 Sistemas de Iluminação pública inteligentes

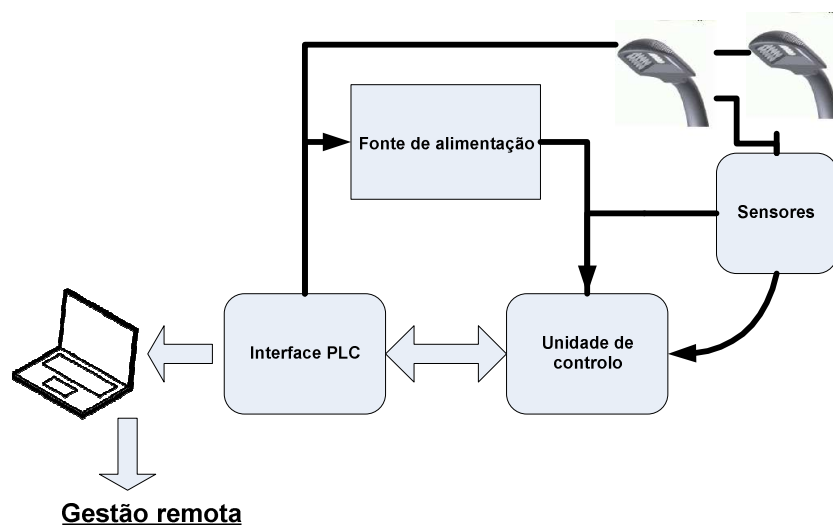
A iluminação pública é essencial para a qualidade de vida das pessoas, desempenhando um importante papel no que diz respeito à diminuição de acidentes e à redução da criminalidade. Assim e uma vez que esta representa cerca de 20% do consumo elétrico total [14] e cerca de 40% dos gastos em eletricidade de uma cidade [15], é necessário encontrar sistemas que maximizem a eficiência energética, uma vez que atualmente existem vários gastos energéticos que são desnecessários, principalmente em áreas rurais em que por vezes se verifica a iluminação de um certo espaço durante toda a noite sem quase existir movimento. É então necessário atuar tendo em vista otimizar os gastos energéticos. Nesse âmbito cada vez mais se fala em sistemas de iluminação com controlo inteligente que possam permitir monitorização, controlo e a introdução de uma rede de comunicação no âmbito de uma redução nos custos da eletricidade, redução nos custos de manutenção e emissão de CO<sub>2</sub> como requerido pelo protocolo de Kyoto [14-15] . Sendo assim um sistema deste tipo, para poder efetuar o controlo desejado, incorpora três blocos principais:

- Transdutores;
- Unidade de controlo;
- Rede de comunicação.

Os transdutores são usados para fazer a transdução de medições pertinentes para o sistema, a unidade de controlo é responsável pelo processamento dos valores lidos pelos sensores, pelas tomadas de decisão e actuação no sistema, enquanto a rede de comunicação permite a troca de dados entre os diferentes componentes do sistema e as luminárias. Em



secções posteriores (2.5, 2.6 e 2.7) falar-se-á um pouco mais detalhadamente destes blocos. Na figura 5 é apresentado um diagrama de blocos de um sistema deste tipo.



**Figura 5 – Diagrama genérico de um sistema de iluminação inteligente.**

Com a grande ascensão dos LEDs na iluminação pública, é agora normal associar sistemas deste tipo a luminárias constituídas por LEDs. Também aqui se verifica a vantagem da introdução de LEDs, uma vez que permite um controlo mais robusto do sistema, tendo em conta a facilidade na integração de sensores e facilidade na variação do fluxo luminoso. No entanto se olharmos por outra perspetiva, ou seja, a introdução de “inteligência” em iluminação com LEDs, também se verifica que é de extrema importância, uma vez que, por exemplo a luz de saída de um LED diminui com o aumento de temperatura deste [12], sendo assim é necessário efetuar um controlo de modo a conseguir monitorizar a temperatura na junção do LED e assim potenciar o tempo de vida deste. É então fulcral associar a utilização de LEDs na iluminação com a introdução de um controlo inteligente, como se poderá ver na secção 2.4 em que será demonstrado um estudo que permite observar que a introdução de controlo permite otimizar significativamente a utilização dos LEDs.

Para concluir são citadas, na tabela 4, algumas vantagens e desvantagens destes sistemas na iluminação com LEDs.

| Vantagens   | Desvantagens  |
|---|---|
| <ul style="list-style-type: none"><li>• Monitorização;</li><li>• Controlo da temperatura dos LEDs;</li><li>• Melhoria da eficiência energética;</li><li>• Diminuição dos custos de manutenção;</li><li>• Diminuição dos gastos em eletricidade;</li><li>• Em caso de avaria, facilidade na deteção desta;</li></ul> | <ul style="list-style-type: none"><li>• Custo inicial;</li><li>• Tecnologia/conhecimento;</li></ul> |

Tabela 4 - Vantagens e desvantagens de sistemas inteligentes de iluminação pública com LEDs

## 2.4 Casos de estudo

O objetivo desta secção é demonstrar, que de facto a iluminação pública recorrendo a sistemas com LEDs e com incorporação de inteligência começaram a ser implantados e a ganhar terreno relativamente às tecnologias atuais. Para esse efeito será feita uma breve referência a alguns países que já começam a procurar e implementar este tipo de sistemas, bem como um caso mais específico de iluminação interna, mas que permite ter uma ideia dos ganhos que se obtêm com a utilização deste tipo de sistemas e em que se faz a comparação entre três sistemas: lâmpadas fluorescentes, LED, LED com controlo inteligente.

### **I. Primeira autoestrada na Alemanha com iluminação utilizando LEDs:**

Segundo notícia avançada pela Led Professional magazine, a 7 de outubro de 2009 [16], na Alemanha, deu-se início a um projeto-piloto na autoestrada A24 em Walsleben que empregava pela primeira vez iluminação numa autoestrada utilizando LEDs. Foram logo notadas claras diferenças, relativamente à qualidade da luz exibida pelos LEDs, demonstrando uma clara melhoria na iluminação já que zonas escuras e longas sombras deixaram de se notar [16].

### **II. Cidades Portugueses expandem a instalação de LEDs na iluminação das estradas:**

Segundo notícia avançada pela mesma fonte, a 7 de setembro de 2009, também em Portugal se começou a verificar utilização de LEDs na iluminação pública. De facto, em abril de 2009 foram então introduzidas as luminárias constituídas por LEDs na cidade de Pombal, oferecendo uma combinação atrativa de eficiência energética, distribuição

uniforme da luz e baixos custos de manutenção e reparo. Perante isto, durante o verão, outros municípios como Alcochete, Moita, Barreiro, Santa Maria da Feira, Vale de Cambra, Arraiolos, Estarreja, Coimbra e Torres Vedras optaram por iluminar as suas estradas utilizando LEDs.

### ***III. Iluminação Inteligente com LEDs:***

Este estudo foi retirado de um artigo da revista Led Professional Review, em que é feita uma comparação entre uma tradicional iluminação com lâmpadas fluorescentes, uma iluminação com LEDs e por fim, iluminação com LEDs mas com a introdução de “inteligência” [17]. De notar, que neste artigo é retratada iluminação interior, no entanto é passível de serem verificadas as melhorias com a introdução do LED e controlo. Sendo assim nas tabelas 5 e 6 são mostrados os resultados obtidos.

Para concluir, são facilmente observáveis nas linhas a sombreado nas tabelas 5 e 6 as reduções económicas e energéticas que a adoção deste tipo de sistemas implicam. De referir a melhoria apresentada pela integração de controlo inteligente que, neste caso, passou pela integração de sensores (luminosidade e movimento) e de um calendário. As regras aplicadas no âmbito do bom funcionamento dos sensores não vão ser aqui descritas, uma vez que o objetivo deste exemplo é apenas demonstrar o aumento de eficiência com a introdução destas soluções.

## **2.5 Sistemas de controlo**

Um sistema de controlo é algo que, assumindo uma determinada entrada vai reagir, atuando sobre a entrada, de modo a obter uma saída cujo resultado seja o esperado. No caso de sistemas de iluminação, essas entradas serão referentes a valores atmosféricos, a presença de movimento, valores de temperatura dos LEDs, entre outras, e é esperado que o sistema atue, conforme o desejado. De seguida serão apresentadas as arquiteturas de controlo, quais as opções de controlo existentes e as respetivas topologias de controlo no âmbito de sistemas de iluminação. Assim, o controlo deve ser diferente consoante o local que se esteja a considerar, conforme as suas principais necessidades e características.

## Comunicação em sistemas de iluminação pública inteligentes

| Trigger                                    | Ação       | Contribuição na poupança de energia |
|--|------------|-------------------------------------|
| Calendário                                 | ON/OFF     | 15%                                 |
| Sensor de luminosidade ambiente            | ON/OFF     | 2%                                  |
|  | Dimming    | 8%                                  |
| Sensor de movimento                        | ON/OFF     | 10%                                 |
|  | Dimming    | 5%                                  |
| <b>Redução total no consumo de energia</b> | <b>40%</b> |                                     |

Tabela 5 [17] - Redução no consumo de energia com o uso de controlo inteligente

| Item   | Fluorescente (CFL) | LED               | LED com controlo inteligente |
|--|--------------------|-------------------|------------------------------|
| Número de lâmpadas/LEDs  | 1                  | 9                 | 9                            |
| Potência média por lâmpada/LED   | 24.00              | 1.03              | 0.62                         |
| Potência média por luminária   | 24.00              | 9.26              | 5.56                         |
| Perdas de potência no balastro ou LED driver                                 | 15%                |                   |                              |
| Watts por luminária (com perdas)   | 24.00              | 9.26              | 5.56                         |
| Tempo de vida médio (horas)  | 2,000              | 60,000            | 60,000                       |
| Consumo de energia (Kwh de eletricidade usada por mais 60k horas)            | 1,440.00           | 556.56            | 333.72                       |
| Taxa de energia  | 0.12               |                   |                              |
| Custos energéticos   | \$172.80           | \$66.68           | \$40.05                      |
| Número de falhas esperadas de CFL ou LED por luminária (depois de 60k horas) | 3                  | 0                 | 0                            |
| Custos totais de potência por luminária (mais de 60k horas)                  | \$172.80           | \$66.68           | \$40.05                      |
| Número de luminárias num quarto  | 25                 |                   |                              |
| Custos totais de energia de todas as estruturas                              | \$4,320.00         | \$1,666.98        | \$1,001.16                   |
| <b>Poupança energética em \$</b>   | <b>-&gt;</b>       | <b>\$2,653.02</b> | <b>\$3,318.84</b>            |
| <b>Poupança energética em %</b>  | <b>-&gt;</b>       | <b>61.4%</b>      | <b>76.8%</b>                 |

Tabela 6 [17] - Potência consumida e comparação de custos

### 2.5.1 Unidade de controlo

Antes de iniciar o estudo às diversas topologias de controlo irá ser feito um breve estudo da unidade de controlo. Assim a unidade de controlo passa por um microcontrolador como por exemplo o PIC18F258. Este será responsável pela leitura, interpretação e posterior transmissão dos valores fornecidos pelos diversos sensores. Para isto, o microcontrolador possui um conjunto de características, tais como [18].

- Memória flash de 32k.
- 22 entradas/saídas.
- *Timers* de 16 bits.
- ADC de 5 canais com uma resolução de 10 bits.
- Modulo CCP (*Capture/Compare/PWM*).

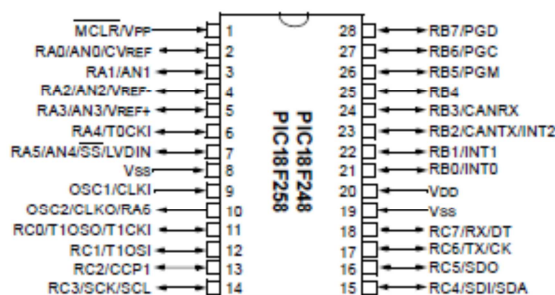


Figura 6 – Esquema dos pinos do PIC18F258.

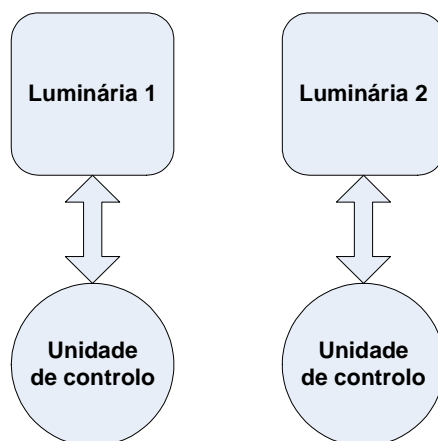
De notar que existe um conjunto de outros microcontroladores que também são baratos e podem ser facilmente incorporados em sistemas deste tipo.

### 2.5.2 Arquiteturas de controlo

Conforme a diversidade de locais, pode ser adequado escolher diferentes arquiteturas de controlo, de acordo com as necessidades. Estas dependem, essencialmente, da forma como se integra a unidade de controlo com o restante sistema de iluminação.

#### ❖ Controlo autónomo:

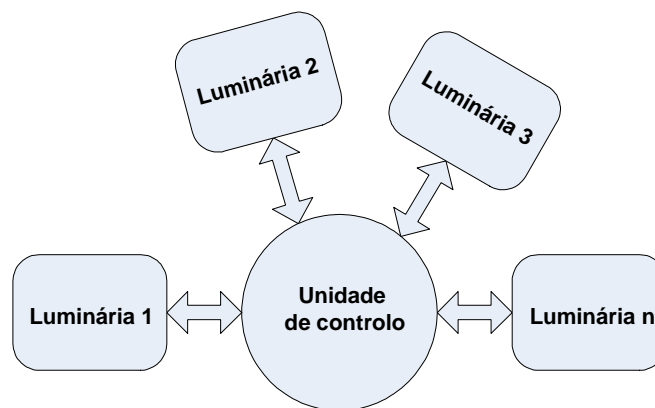
É uma solução em que cada luminária possui uma unidade de controlo própria, permitindo que cada ponto de iluminação seja gerido de forma independente. Na figura 7 pode-se observar um controlo deste tipo.



**Figura 7 - Controlo autónomo.**

❖ **Controlo centralizado:**

Aqui a gestão é feita de uma forma centralizada, como o nome indica, ou seja o controlo é feito por uma unidade local que é responsável por um grupo de luminárias. Com esta solução verifica-se um menor custo dos sistemas uma vez que apenas dispomos de uma unidade de controlo. Na figura 8 pode-se observar um controlo deste tipo.



**Figura 8 - Controlo centralizado.**

❖ **Gestão remota:**

Sistemas que dispõem de uma unidade de controlo local que, através da criação de uma comunicação adequada, pode trocar dados com uma estação de controlo remota, podendo assim ser acedida para configurações, sem que seja necessário a deslocação ao local. Na figura 9 pode-se observar esta configuração.

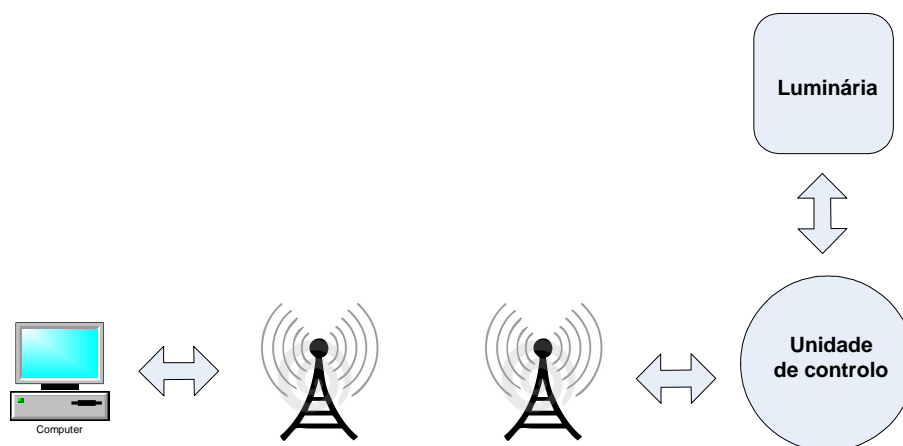


Figura 9 - Gestão remota.

### 2.5.3 Opções de controlo

De igual importância são as opções de controlo que se podem efetuar de maneira diferente, consoante as necessidades do meio circundante. Assim existem duas opções, sendo elas:

❖ **Controlo passivo:**

Sistema de iluminação pública em que o controlo das luminárias é feito, simplesmente, por sensores de luminosidade ambiente ou tendo em conta um horário pré-estabelecido o que acontece, atualmente, em praticamente todos os sistemas de iluminação pública. Sistema simples e barato que pode ser útil em áreas urbanas, onde não há grande necessidade de controlo sobre as luminárias e o seu fluxo luminoso, uma vez que são áreas bastante movimentadas em que é sempre necessário máximo fluxo.

❖ **Controlo ativo:**

Estes são sistemas em que é possível fazer uma gestão da energia, fazendo uso de várias informações de entrada (como a existência ou não de movimento) e decidindo de uma forma ativa, quando e durante quanto tempo, por exemplo, as luminárias se mantêm em máximo fluxo. São sistemas mais caros mas que poderão ser bastante úteis em áreas rurais onde se verificam grandes gastos energéticos, sendo estes desnecessários.

### 2.5.4 Topologias de controlo

Com as arquiteturas e opções definidas anteriormente é possível integrá-las, de forma a desenvolver sistemas que podem funcionar de formas distintas. Existem assim 4 topologias que podem ser consideradas.

### ❖ Topologia de um sistema autónomo com controlo passivo de energia:

Numa topologia deste tipo cada luminária irá dispor de uma unidade de controlo, em que o controlo é feito, basicamente, através de um sensor de luminosidade ou através de um horário predeterminado, ou seja, as luminárias ligam e desligam consoante uma hora desejada. De notar que como o controlo é passivo, entenda-se como unidade de controlo o sensor de luminosidade. Na figura 10 ilustra-se uma topologia deste género.

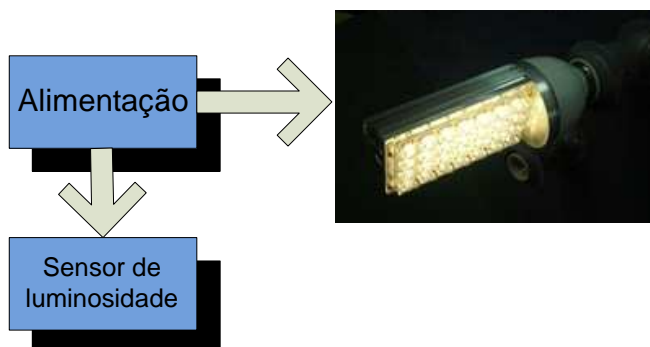


Figura 10 – Topologia de um sistema autónomo com controlo passivo de energia.

### ❖ Topologia de um sistema autónomo com controlo ativo de energia:

Numa topologia deste tipo já é feito um controlo ativo da energia, ou seja, é incorporada uma unidade de controlo que processa um conjunto de dados relativos ao sistema (proveniente de um conjunto de sensores) e que opera de acordo com o esperado. De referir que, como o sistema é autónomo, cada luminária irá possuir uma unidade de controlo. Na figura 11 é exemplificado um sistema deste tipo.

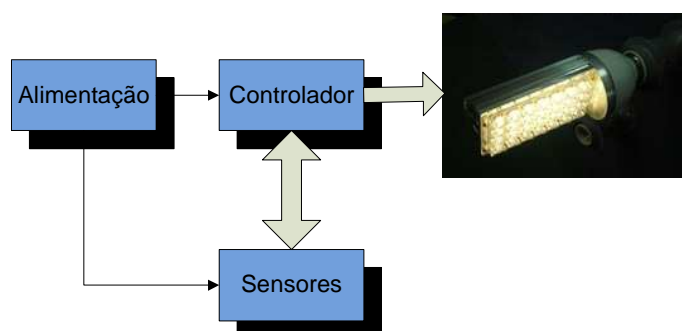


Figura 11 – Topologia de um sistema autónomo com controlo ativo de energia.



### ❖ Topologia de um sistema centralizado com controlo passivo de energia:

Nesta topologia, irá existir, por exemplo, um sensor de luminosidade que será utilizado por um conjunto de luminárias, determinando o estado destas, como se pode observar na figura 12.

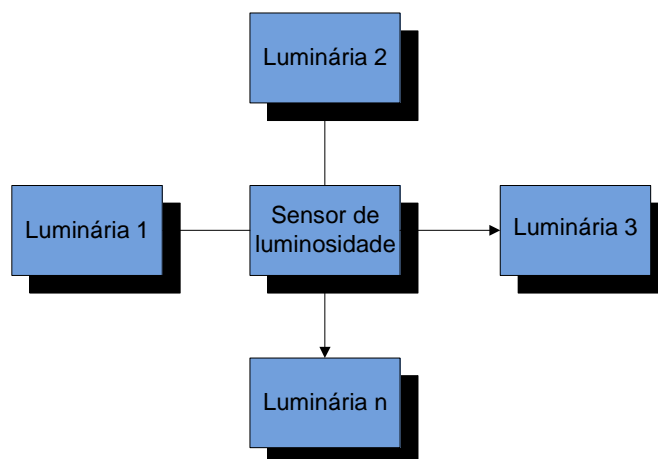


Figura 12 – Topologia de um sistema centralizado com controlo passivo de energia.

### ❖ Topologia de um sistema centralizado com controlo ativo de energia:

A topologia representada na figura 13 é idêntica à topologia ilustrada na figura 11, com a diferença de que existe apenas uma unidade de controlo que irá ser responsável por um conjunto de luminárias, processando os valores provenientes dos sensores.

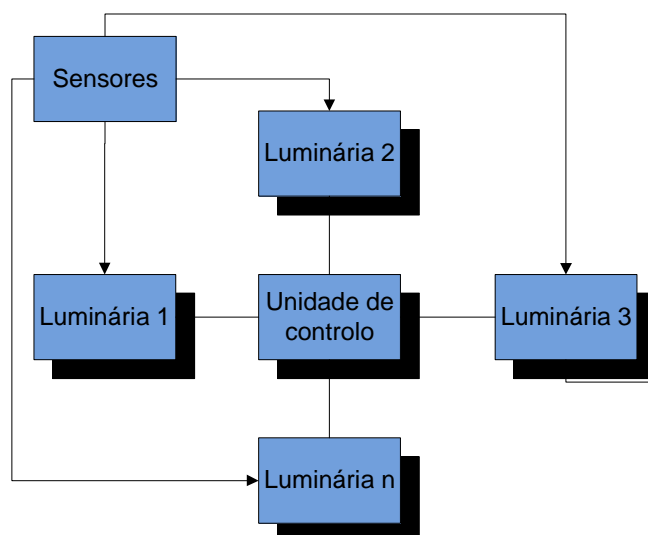
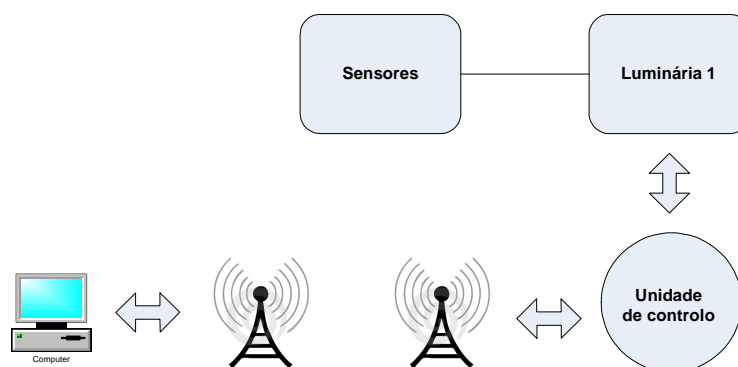


Figura 13 – Topologia de um sistema centralizado com controlo ativo de energia.

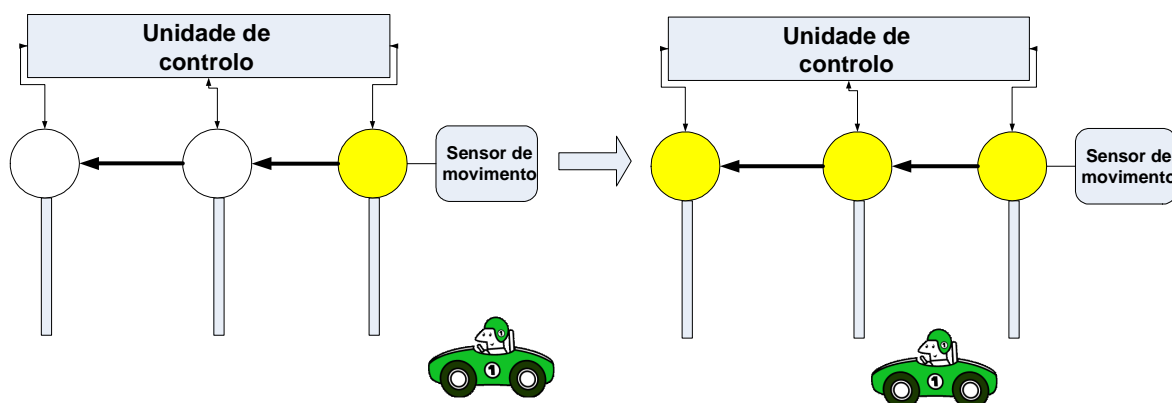
### 2.5.5 Gestão remota

No que diz respeito à gestão remota, esta pode ser aplicada a qualquer tipo de topologia, criando, como já foi dito, uma comunicação (será abordado na secção 2.7) entre a unidade de controlo local/unidades de controlo individuais (sistema centralizado e sistema autónomo, respetivamente) e a entidade responsável pela gerência deste sistema. De notar que, para o caso de um sistema de controlo autónomo, esta opção acaba por se tornar pouco rentável, uma vez que cada unidade ficaria muito complexa.



**Figura 14 – Exemplo de um sistema utilizando gestão remota.**

Na figura 14 é possível observar-se, apenas como exemplo, um sistema autónomo com controlo ativo de energia que emprega gestão remota, ou seja é incorporada comunicação entre os diferentes blocos do sistema de modo a que seja feita a transferência de informação entre estes. Sendo assim é possível monitorizar o sistema à distância. A introdução de comunicação é então bastante importante uma vez que permite diminuir os custos de manutenção, bem como diminuir os custos do sistema. Para esse efeito, pode-se optar por uma estratégia que passa por incorporar apenas um sensor numa luminária e utilizar a rede de comunicação para que as restantes luminárias troquem dados entre si e assim atualizarem o seu estado conforme o esperado. Se para alguns sensores não se torna necessário utilizar esta estratégia, uma vez que o seu custo não o justifica, para outro tipo de sensores, como por exemplo de nevoeiro, (que podem ser pertinentes para alguns locais) e uma vez que estes apresentam um custo bastante elevado, não se torna rentável incorporar um sensor em cada uma das luminárias. Assim uma opção, passa por instalar apenas um sensor destes numa luminária só e utilizar a rede de comunicação para que este sensor seja comum a um conjunto de luminárias.



**Figura 15 – Ilustração da utilização de um sensor para controlar um conjunto de luminárias.**

Na figura 15 está ilustrado o que foi dito acima, no caso um sensor de movimento. É apenas incorporado na primeira luminária e quando este deteta movimento no seu raio de ação, a respetiva luminária acende e envia essa informação às restantes, fazendo com que estas liguem. Alguns aspetos têm que ser tidos em conta, principalmente no caso em que, quando o veículo sair do raio de deteção do sensor, as restantes luminárias não podem desligar imediatamente, logo terá que ser feito o devido controlo do sistema. Neste exemplo o sensor apenas controla três luminárias, mas poderá controlar um conjunto maior e, nesse caso, poderão ocorrer falhas senão se atuar de uma forma correta. Assim, quando o sensor não detetar mais movimento, a luminária não poderá enviar logo essa informação às restantes, uma vez que poderá ocorrer o caso em que estas desliguem estando ainda o veículo a percorrê-las. Como tal, será necessário incorporar um tempo de espera predeterminado (e de acordo com o respetivo local e conjunto de luminárias controladas) para que a luminária “principal” envie a informação para que as restantes luminárias desliguem. Esse tempo poderá ser facilmente calculado, sabendo a distância que as luminárias distam e utilizando uma velocidade média que, de forma a evitar falhas, seja suficientemente lenta. De notar que, alguns aspectos têm que ter tido em conta como, por exemplo, o atraso de propagação dos dados.

## 2.6 Sensores e variáveis de medida

Para fazer uma gestão da energia, um conjunto de valores/variáveis devem ser medidos para que se possa atuar de forma correta. Para isso são então utilizados sensores que fazem a medição de parâmetros do sistema (como a temperatura dos LEDs) convertendo esse valor numa grandeza elétrica que possa facilmente ser processada por

uma unidade de controlo. Como já foi dito anteriormente, a utilização de LEDs em sistemas de iluminação pública facilita bastante a integração de sensores e assim sendo há que fazer uso desta vantagem para tornar os sistemas inteligentes.

No âmbito dos sistemas de iluminação pública podemos classificar três grupos principais de sensores consoante os parâmetros que medem e a funcionalidade que estes têm para o sistema. Na figura 16 pode-se observar um esquema identificando esses grupos. De referir que nas subsecções seguintes apenas é feita uma descrição de variáveis importantes para o sistema, e são identificados sensores ou processos adequados para a medição destas.

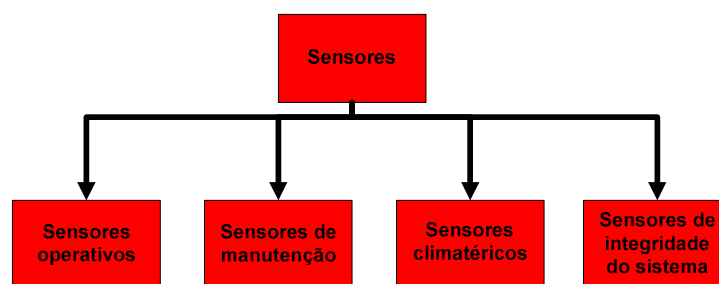


Figura 16 – Classificação dos sensores.

### 2.6.1 Sensores Operativos

Uma das características mais importantes num sistema de iluminação é o controlo do fluxo luminoso. Para isso os sensores operativos são de extrema importância, pois estes permitem o controlo direto do fluxo luminoso. Para esse efeito, é necessário fazer a medida de duas variáveis importantíssimas: a luminosidade ambiente e a existência de movimento.

- **Sensores de luminosidade ambiente**

A luminosidade ambiente é uma importante medida em sistemas de iluminação pública. Assim como se deseja que as luminárias liguem quando estritamente necessário, o sensor estará continuamente a dar informações acerca do nível de luminosidade logo, quando este descer a um certo nível preestabelecido, as luminárias irão ligar de maneira a esses mesmos níveis de luminosidade serem atingidos. Por outro lado, enquanto a luminosidade ambiente estiver acima desse nível, estas irão permanecer desligadas. Apenas de referir que a unidade usada em fotometria para medir luminosidade ambiente é a iluminância (E). Esta mede o fluxo luminoso total por unidade de área [19].

$$E = \frac{dF}{dA}$$

Para este efeito existem sensores que são facilmente realizáveis, sendo eles:

- Fotoresistências;
- Fotodíodos;
- Fototransístores.

### • Movimento

A detecção de movimento é uma das principais medidas no âmbito da eficiência energética uma vez, que se não for detetado nenhum movimento, não se torna necessário que as luminárias estejam ligadas.

Para a detecção de movimento existem um conjunto de sensores, completamente realizáveis [19-20]:

- Detetores de humidade, vibração e stress;
- Detetores acústicos;
- Detetores de movimento com vídeo;
- Detetores capacitivos;
- Detetores triboelétricos ;
- Detetores ultrasónicos e micro-ondas;
- Detetores optoelectrónicos;
- Detetores de movimento sensíveis a radiação na gama do infravermelho.

## 2.6.2 Sensores de Manutenção

Como o nome indica são sensores que têm como principal função garantir a manutenção e o bom funcionamento do sistema. Estes medem valores pertinentes do sistema, como a temperatura dos LEDs, sendo estes valores processados pela unidade de controlo e, caso seja necessário, alguma medida será efetuada (baixar, por exemplo, a corrente nos LEDs caso a temperatura destes seja elevada).

### • Temperatura dos LEDs

Como já foi dito, o controlo da temperatura dos LEDs é importante, de modo a maximizar o tempo de vida dos dispositivos. A unidade do SI para a temperatura é o

Kelvin, sendo definido como "*a fração (1/273,16) da temperatura termodinâmica do ponto triplo da água*". No entanto existem outras escalas para medir temperatura, sendo as mais utilizadas, Celsius (°C) e Fahrenheit (°F). Facilmente se passa duma unidade para as outras, como se mostra de seguida:

$$\text{Graus Celsius} = \text{Graus Kelvin} - 273.15$$

$$\text{Graus Celsius} = (\text{Graus Fahrenheit} - 32) 1.8$$

Para ser feita a sua medida existem vários sensores, como:

- Termopar;
  - Sensores de resistência térmica (RTDs);
  - Termístor;
  - Sensores integrados.
  - Junções p-n.
- 
- **Corrente e tensão**

Um dos grandes objetivos destes novos sistemas de iluminação é diminuir o consumo de potência, pelo que se torna então necessário seguir continuamente a potência consumida pelo sistema. Para isso é então fulcral tirar medidas da tensão (V) e corrente (I) fornecida à luminária, permitindo verificar e corrigir alguma anomalia que ocorra. Também para melhorar o consumo de potência é necessário medir e corrigir o fator de potência ( $\cos(\gamma)$ ), que é dado por [19]:

$$P_{\text{activa}} = V_{\text{ef}} \cdot I_{\text{ef}} \cdot \cos \gamma$$

A unidade de potência vezes unidade de tempo é, usualmente, o KWh. Facilmente se conseguem tirar medidas acerca da corrente e tensão que está a ser fornecida aos LEDs, utilizando circuitos eletrónicos apropriados (como um simples divisor resistivo) e devidamente integradas com um micro-controlador.

### 2.6.3 Sensores Climatéricos

Este grupo pode englobar um vasto número de sensores consoante as condições climatéricas. Estes podem ser de extrema importância na regulação das luminárias em

condições climatéricas adversas como chuva intensa, nevoeiro ou neve, no sentido de proporcionar maior segurança aos condutores. Estes devem ser utilizados tendo em conta as condições climatéricas da região em que se está a considerar instalar o sistema.

- **Chuva, neve e nevoeiro**

Chuva, neve e nevoeiro serão, porventura, as condições climatéricas mais importantes a ter em conta em sistemas de iluminação uma vez que em condições adversas de chuva, neve ou nevoeiro a visibilidade é reduzida, podendo ser necessário executar um ajuste no fluxo luminoso das luminárias para assim auxiliar os condutores e proporcionar maior segurança. Para isso podem ser usados sensores que detetem estas condições.

### **2.6.4 Sensores de Integridade do sistema**

São sensores que têm como função verificar a integridade do sistema, ou seja, verificar e informar se alguma estrutura do sistema está danificada como, por exemplo, um poste de iluminação.

- **Estrutura de iluminação**

Em caso de queda de alguma das estruturas de iluminação seria pertinente que a entidade responsável pela manutenção do sistema fosse imediatamente advertida. Assim, uma solução possível seria munir os postes de iluminação com sensores de deslocamento de maneira a ser detetado o deslocamento da estrutura, caso ocorra. De notar que apenas fará sentido utilizar estes sensores se o sistema possuir algum tipo de comunicação. No entanto para este fim são, normalmente, usados os chamados sensores de *tilt*. Estes podem consistir num pêndulo em que, existindo inclinação da estrutura, o contacto é fechado, advertindo os serviços de manutenção que existiu movimento do poste. De seguida são enunciados alguns sensores de deslocamento [20]:

- Potenciómetros;
- Sensores capacitivos;
- Sensores indutivos e magnéticos (LVDT, RVDT);
- Sensores óticos.

### **2.7 Integração de comunicação**

Como já foi dito, uma das maneiras de maximizar a eficiência dos sistemas de iluminação, bem como minimizar os custos de manutenção, é integrar nestes comunicação que poderá ser feita entre os componentes de controlo e as luminárias, bem como uma estação remota que possa controlar o sistema à distância. O grande objetivo desta secção é apenas demonstrar a importância de possuir uma rede de comunicações devidamente projetada no âmbito de sistemas de iluminação pública. No capítulo seguinte será feito um estudo mais aprofundado acerca das diversas considerações a serem feitas e das diferentes opções possíveis para a sua devida implementação.

Deste modo a integração de comunicação é um dos aspetos fulcrais para tornar estes sistemas inteligentes, existindo já, vários softwares desenvolvidos para a gerência destes sistemas [21]. Assim, como descrito na secção anterior, se a utilização de sensores que possibilitem a medida de certas variáveis importantes para o sistema, é de extrema importância, a integração de uma rede que permita que as luminárias troquem esses mesmos valores entre si e uma entidade de supervisão é uma funcionalidade que permite otimizar bastante estes sistemas, sobretudo a nível da manutenção do sistema. Os dados provenientes dos sensores serão os principais responsáveis de como é feita a gestão do sistema. Se por exemplo uma luminária estiver equipada com um sensor de integridade do sistema e se for detetada a inclinação da estrutura de iluminação, é de todo pertinente que seja logo enviada uma mensagem de aviso para a entidade responsável pela gestão do sistema. O mesmo se passa se, por exemplo, for detetada uma temperatura muito elevada para os LEDs. Verifica-se assim que a interligação dos dados sensoriais com uma rede de comunicação é de extrema importância para que se possa maximizar a eficiência destes sistemas. Na figura 17 está representado um diagrama de blocos referente a um sistema possuindo comunicação.

Por outro lado, como já se demonstrou na figura 15, a integração de uma rede de comunicação pode permitir também a partilha de dados sensoriais entre um grupo de luminárias como, por exemplo, informação acerca da luminosidade, diminuindo assim o custo pela redução de alguns sensores. Existe assim um vasto grupo de opções que podem ser exploradas em sistemas deste tipo. Mesmo variáveis que digam respeito à manutenção de cada luminária podem ser utilizadas para melhorar o desempenho do sistema. Por exemplo, se for detetado que uma luminária foi danificada, ou que vários LEDs falharam,



estando esta a funcionar anormalmente faz todo o sentido partilhar esta informação com a entidade que gere o sistema, para que se aumente o fluxo luminoso das luminárias que estão em torno daquela que está danificada para que se possa manter minimamente a uniformidade luminosa.

Como já foi dito o objetivo desta secção era apenas demonstrar a importância da integração de comunicação nestes sistemas. No capítulo seguinte será feito um estudo mais aprofundado acerca dos passos e escolhas a serem feitas para implementar comunicação entre os diferentes blocos. Para finalizar apresentam-se na tabela 7 algumas vantagens e desvantagens identificadas.

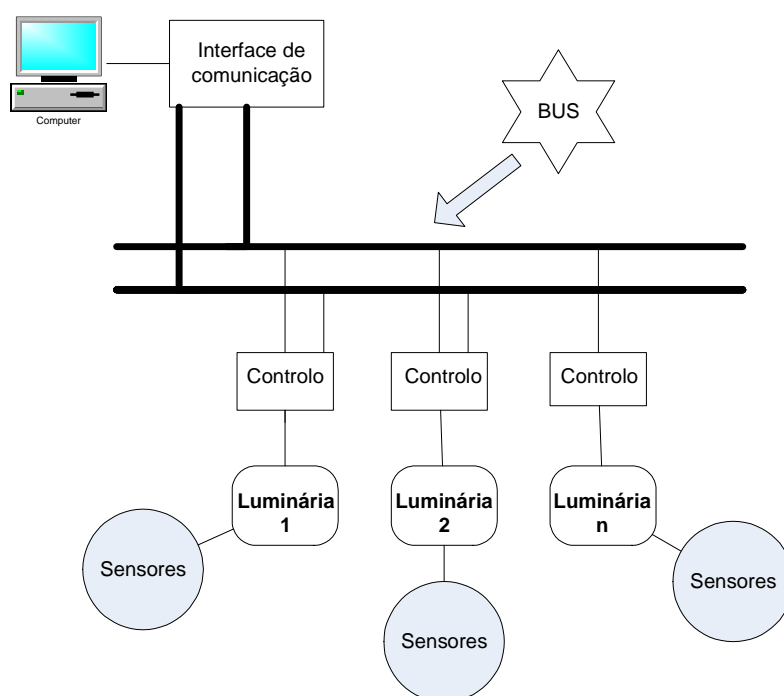


Figura 17 – Diagrama de blocos de um sistema com comunicação.

| Integração de comunicação  |   |
|--|---|
| Vantagens  | Desvantagens  |
| <ul style="list-style-type: none"><li>• Manutenção e controlo do sistema à distância.</li><li>• Aumento da eficiência.</li><li>• Partilha de dados sensoriais.</li><li>• Diminuição do número de sensores.</li></ul> | <ul style="list-style-type: none"><li>• Sistema eletronicamente mais “pesado”.</li><li>• Custo inicial.</li></ul> |

Tabela 7 – Vantagens e desvantagens da integração de comunicação

### **2.8 Em resumo**

A iluminação pública desde sempre foi um importante fator para a sociedade. Esta foi evoluindo até que atualmente as tecnologias de iluminação mais utilizadas são as lâmpadas de vapor de sódio e as de iodetos metálicos. As lâmpadas de vapor de sódio podem-se dividir em duas: as de alta pressão e as de baixa pressão. As de baixa pressão apresentam grandes eficiências mas possuem um IRC muito baixo. As de alta pressão já apresentam um IRC aceitável, no entanto apresentam um arranque lento. Por sua vez as lâmpadas de iodetos metálicos apresentam uma eficiência energética bastante boa mas apresentam um custo inicial relativamente alto.

Atualmente emergiu a tecnologia LED, com o aparecimento do LED de luz branca. Estes apresentam características muito boas, possuindo um IRC bom, facilidade na variação do fluxo luminoso e facilidade na integração de eletrônica auxiliar. Permitem assim obter grandes eficiências energéticas. Todas estas características permitem tornar os sistemas cada vez mais inteligentes, introduzindo nestes: controlo, sensores e comunicação. Com sistemas destes pode haver reduções na ordem dos 78% no consumo de energia.

Existem várias técnicas para proceder ao controlo destes sistemas, resumindo-se estas a quatro: sistemas autónomos com controlo passivo de energia, sistemas autónomos com controlo ativo de energia, sistemas centralizados com controlo passivo de energia e sistemas centralizados com controlo ativo de energia. A nível sensorial, grandezas como temperatura dos LEDs, corrente e tensão fornecida aos LEDs, ocorrência de movimento e condições climatéricas são bastante pertinentes para que se possa tornar estes sistemas mais eficientes. Para complementar pode ser introduzida comunicação que permita fazer a troca de dados entre o sistema e a entidade que o regula, de modo a ser possível configurar o sistema à distância.

# **Capítulo III**

## **3 Redes de comunicação**

No capítulo anterior ficou explícita a importância da integração, não só da lâmpada LED, mas também de todo um conjunto de blocos que permitam tornar os sistemas de iluminação pública inteligentes. Assim, a melhor forma de fazer a interligação dos diferentes blocos é a introdução de uma rede de comunicações, permitindo também fazer a gestão do sistema a partir de um terminal remoto.

No presente capítulo é então objetivo fazer um estudo acerca de redes de telecomunicações e das diferentes opções que estão disponíveis para a sua implementação. Para isso começa-se por abordar e explicar várias topologias de rede que podem ser adotadas. Seguidamente será feito um estudo acerca das arquiteturas de rede que podem ser implementadas. Serão também apresentadas algumas opções para o meio físico a utilizar para a transferência de dados. Por fim faz-se um estudo acerca do interface de comunicação que se pode implementar (porventura o mais usado) ao nível do microcontrolador.

### 3.1 Topologias de rede

A topologia de rede descreve como estão organizados os diferentes dispositivos na rede, física e logicamente. Ou seja fisicamente refere-se ao aspeto da rede e logicamente refere-se a como é feito o fluxo de dados através da rede. Existem assim várias topologias de rede que podem ser adotadas. Seguidamente é feita uma descrição de cada uma das topologias existentes.

#### ❖ Topologia em barramento

É uma topologia em que todos os nós estão ligados a um mesmo barramento físico de dados e sendo assim pode-se dizer que o barramento é partilhado. Nesta topologia apenas um nó pode “escrever” no barramento num dado instante. Os restantes “escutam” e recebem os dados (destinados a si). É uma topologia que já não é muito utilizada fisicamente, no entanto logicamente ainda é bastante utilizada. Na figura 18 pode-se observar um esquema desta topologia.

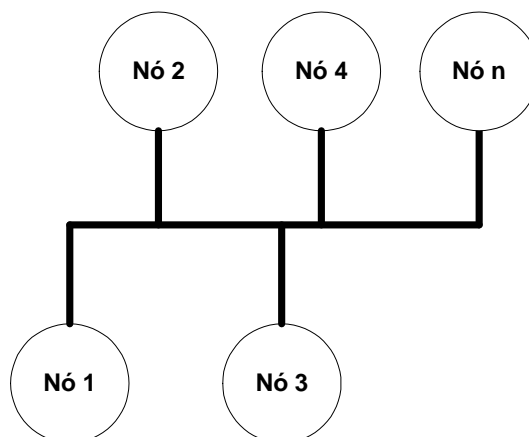


Figura 18 – Topologia de rede em barramento.

#### ❖ Topologia ponto-a-ponto

Nesta topologia é criada uma ligação direta entre os nós da rede, permitindo uma comunicação simples e direta. Na figura 19 pode-se observar um esquema desta topologia.



Figura 19 – Topologia ponto-a-ponto.

### ❖ Topologia em estrela

A topologia em estrela consiste num *hub* central que é ligado aos restantes nós da rede usando uma ligação ponto-a-ponto. A grande vantagem desta topologia consiste na facilidade em integrar novos nós à rede, contudo uma falha no *hub* central implica uma falha total da rede, como se pode verificar na figura 20.

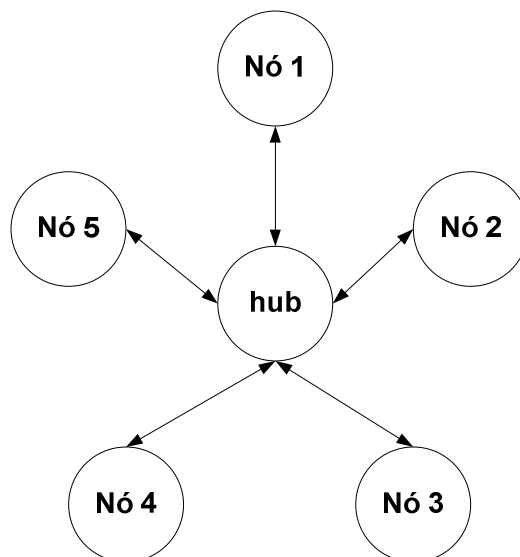


Figura 20 – Topologia de rede em estrela.

### ❖ Topologia em anel

É uma topologia em que os nós são distribuídos numa forma fechada, onde os dados circulam entre nós numa dada direção. Cada dispositivo pertencente a esta rede irá funcionar como um repetidor de modo a manter a força do sinal. Assim, cada dispositivo incorpora um recetor para o sinal de entrada e um transmissor para enviar o sinal para o próximo nó da rede. A rede está assim dependente da capacidade do sinal viajar em torno do anel [22], sendo que, a falha de um dispositivo implica a falha de toda a rede. Na figura 21 pode-se observar um esquema com esta topologia.

### ❖ Topologia em árvore

Esta também pode ser designada como rede hierárquica. Existe um nó central que está no topo da rede (no topo da hierarquia) e que vai ligar-se a um ou mais nós que farão parte do segundo nível da hierarquia. De notar que as ligações entre o nó central e os nós do segundo nível são ponto-a-ponto. Por sua vez, os nós do segundo nível vão ligar-se a

outros nós que irão formar o terceiro nível e assim sucessivamente, usando sempre ligações ponto-a-ponto. Esta distribuição pode ser vista na figura 22.

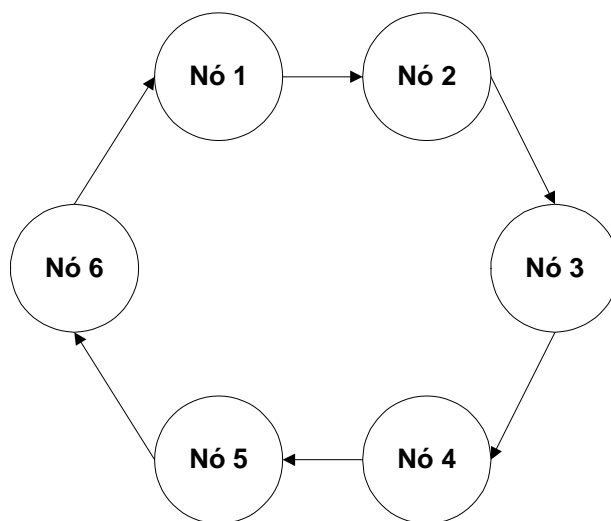


Figura 21 – Topologia de rede em anel.

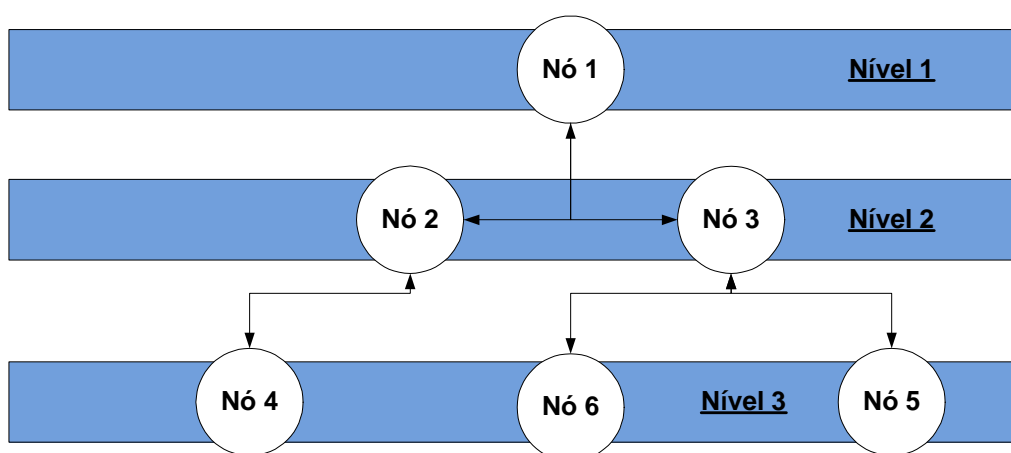


Figura 22 – Topologia de rede em árvore.

### ❖ Topologia em malha

Esta é a topologia mais importante para o controlo da iluminação [17]. Nesta topologia todos os nós estão interligados, fazendo com que a rede seja flexível e redundante. A grande vantagem desta rede é que todos os nós são considerados como pares equivalentes, dando-lhes capacidade para se coordenarem entre si e oferecendo à rede possibilidade de se “autocurar”. Se um nó falhar, outro nó irá encaminhar os dados para o nó destino. Esta propriedade pode ser observada na figura 23 [17].

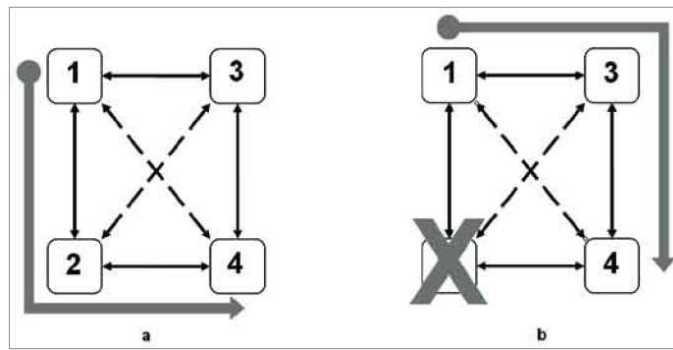


Figura 23 – Exemplo de uma rede em malha e da sua robustez [17].

### 3.2 Arquiteturas de rede

Uma vez feito o estudo acerca das topologias que uma rede pode assumir, é de igual modo importante saber como estão organizados logicamente os diferentes dispositivos da rede, ou seja, como é controlada a transferência de dados. Para isso uma rede pode assumir diferentes arquiteturas que são discutidas de seguida.

#### ○ Arquitetura Master-Slave

Como o nome indica, nesta configuração existe um dispositivo que é considerado como *master* e controla todas as trocas entre os dispositivos da rede. Este também será responsável pela sincronização da rede. Os *slaves* podem trocar dados entre eles, de acordo com as especificações do *master* (dependendo do protocolo de comunicação utilizado). Na seguinte figura pode-se observar uma arquitetura deste tipo.

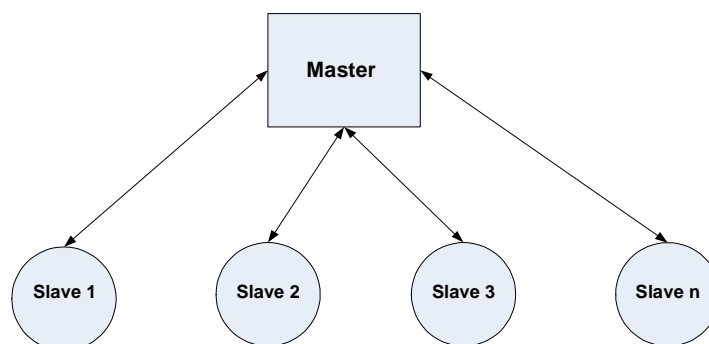


Figura 24 – Arquitetura *master-slave*.

#### ○ Arquitetura Master-Slave dinâmica

A grande diferença desta arquitetura relativamente à arquitetura *master-slave* convencional é que, neste esquema, o primeiro nó a iniciar uma transmissão será considerado como o *master* [23]. Daí a referência a uma arquitetura dinâmica.

### ○ Arquitetura multimaster

Como o nome indica é uma arquitetura em que todos os dispositivos da rede são considerados como *masters*. Sendo assim, qualquer dispositivo pode iniciar uma transferência de dados, podendo haver transferências entre todos os dispositivos da rede.

### ○ Arquitetura Peer-to-Peer

Nesta configuração todos os dispositivos da rede desempenham o mesmo papel, ou seja, estão todos num nível hierárquico idêntico. Assim podem ocorrer trocas entre os dispositivos sem que haja controlo de um *master*.

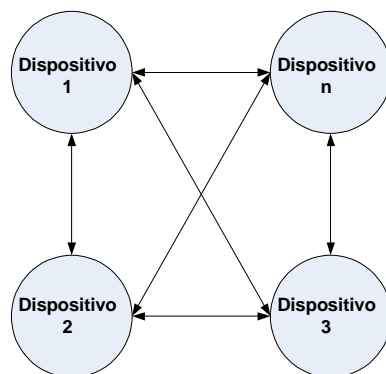


Figura 25 – Arquitetura *peer-to-peer*.

### ○ Arquitetura Centralizada

Este é uma mistura entre a arquitetura *master-slave* e a arquitetura *peer-to-peer*, em que existe um dispositivo central que é responsável pela gestão da rede e das trocas entre os dispositivos. Os outros dispositivos podem efetuar trocas entre si sem ter que passar pelo dispositivo central.

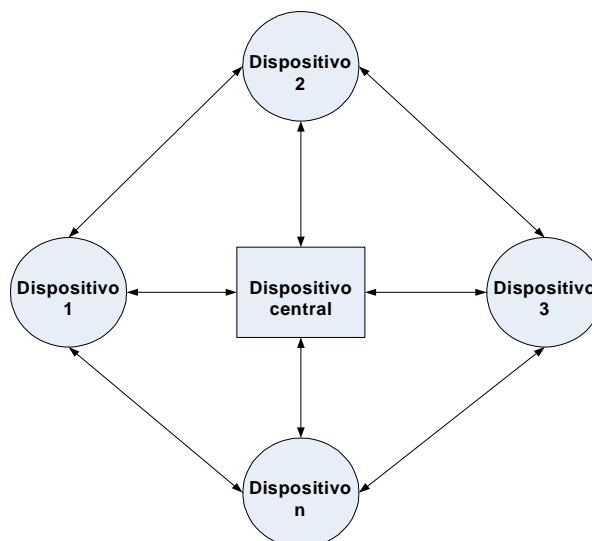


Figura 26 – Arquitetura centralizada.



### 3.3 Meio Físico

Uma vez estudadas as topologias e arquiteturas de rede existentes, é então pertinente falar acerca da camada física por onde irão ser transportados os dados, ou seja, refere-se à camada 1 segundo o modelo OSI (*Open Systems Interconnections*). Neste âmbito serão discutidas nas seguintes subsecções quatro formas de o fazer: Wireless, *Power Line Communications*, RS-484 e RS-232.

#### 3.3.1 Wireless

Um dos aspetos de grande importância na comunicação via Wireless é a banda de frequência utilizada. Nesse âmbito estes sistemas operam, geralmente, à frequência de 2.4 GHz ou utilizam a banda sub-GHz [17]. De notar que diferentes normas operam dentro da banda de 2.4 GHz, como por exemplo, a norma IEEE 802.11, que tem como vantagem o facto de diferentes protocolos desta (WiFi) poderem transmitir dados em diferentes canais de frequência sem que interfiram uns com os outros. A grande desvantagem desta norma é que é desenhada para média distância. Outra norma que utiliza a banda 2.4 GHz é a norma IEEE 802.15.4. Uma tecnologia que a utiliza é, por exemplo, o ZigBee que tem vantagens como fácil adaptação à rede, baixo consumo de potência, comunicação credível e baixo custo [24]. Contudo, facilmente ocorrem interferências com outras tecnologias que utilizem a mesma frequência como por exemplo o WiFi. Uma tecnologia como a ZigBee usa uma topologia em pseudo-malha, como é demonstrado na figura 27 [17]. Como é possível observar na figura, esta topologia é idêntica a uma topologia em malha, no entanto existe um nó, designado como “Coordinator Node” que coordena todas as ações da rede. Assim esta topologia tem como principal desvantagem o facto de que se o nó principal falha, toda a rede irá falhar, o que para sistemas de iluminação inteligentes não é suportável. No entanto existem vários protocolos baseados na norma 802.15.4 que possibilitam redes em malha.

Outra banda de frequência que pode ser utilizada para comunicação wireless é a banda sub-GHz. Esta apresenta algumas vantagens relativamente à banda 2.4 GHz, uma vez que apresenta menores interferências e menor consumo de potência. Permite também a transmissão a longas distâncias devido à fraca atenuação. Uma desvantagem da utilização desta banda de frequência é que é regional, dificultando assim a vida aos fabricantes deste tipo de sistemas, uma vez que se torna complicado implementar uma solução que seja

funcional em qualquer lado do globo. Por exemplo 915 MHz é bastante utilizado na América do Norte, 868 MHz na Europa e outras frequências são utilizadas por toda a Ásia [17].

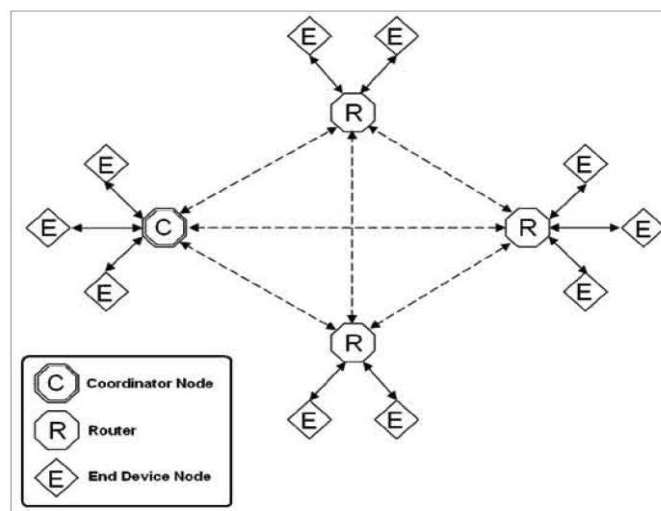


Figura 27 – Exemplo de uma rede ZigBee [17].

### 3.3.2 Power Line Communications

Outra forma de se transportarem os dados numa rede é utilizando a tecnologia *Power Line Communications*.

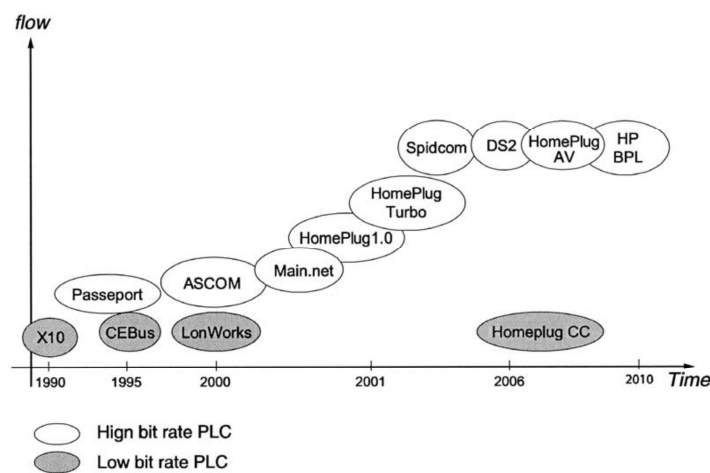
PLC é uma tecnologia de comunicação que faz uso do meio e da rede elétrica de baixa tensão para providenciar serviços de comunicação [25], ou seja permite a transmissão de dados sobre a rede elétrica.

Esta não é uma tecnologia propriamente recente, aparecendo em 1950 o primeiro sistema PLC, em que a frequência da portadora se situava entre 100Hz e 1KHz. Na figura 28 pode ser observado a evolução das tecnologias que utilizam PLC [25].

Como já foi dito, este meio de comunicação permite enviar dados sobre a rede elétrica pública. Assim a camada física segundo o modelo OSI, para esta comunicação serão os cabos elétricos e, de uma forma mais geral a rede elétrica, que têm características próprias que têm que ser tidas em conta. De notar que a principal função destes é transportar energia à tensão e frequência de rede. Por exemplo na Europa a tensão e a frequência da rede pública é de 230V e 50 HZ respetivamente [25]. Assim em sistemas PLC é necessário ter em conta as características do suporte físico, como sejam:

- Impedância;

- Capacitância e Indutância;
- Perturbações e ruído eletromagnético;
- Atenuação;
- Acoplamento entre fases;
- Resposta em frequência;
- Sensibilidade dos interfaces;



**Figura 28 – Evolução das tecnologias PLC [25].**

Assim é necessário fazer uma modulação do suporte físico. De referir que para o transporte do sinal na rede pública é necessário recorrer a um sinal de baixa amplitude devidamente modulado em torno da frequência da portadora. De facto, pode-se dizer que a camada física consiste neste sinal modulado transportado na rede elétrica a uma frequência determinada pela tecnologia PLC [25]. Este sinal pode ser modulado em fase, amplitude ou frequência. São atribuídas duas bandas de frequência para tecnologias PLC [25]:

- 3 a 148 KHz para um baixo bit rate;
- 2 a 20 MHz para um alto bit rate;

Uma das principais funcionalidades de uma rede PLC é o modo como esta está organizada, de modo a controlar todos dispositivos da mesma rede. As tecnologias PLC usam um dos três modos de rede seguintes:

Para finalizar esta subsecção, é demonstrado na tabela 8 um conjunto de vantagens e desvantagens desta tecnologia [25-26].

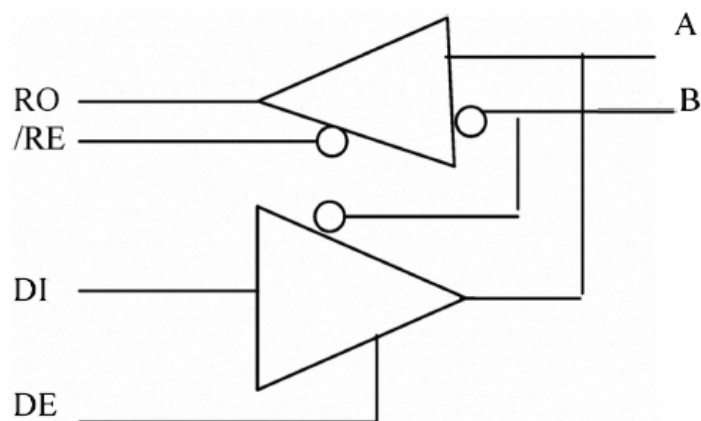
| Vantagens  | Desvantagens  |
|--|---|
| <ul style="list-style-type: none"><li>• Utilização de uma rede elétrica já existente.</li><li>• Desenvolvimento rápido.</li><li>• Método de encriptação robusto.</li><li>• Menores interferências.</li></ul> | <ul style="list-style-type: none"><li>• Custo elevado.</li><li>• Necessidade de vários componentes para realizar a comunicação.</li><li>• Para altas transmissões, problemas relacionados com compatibilidade eletromagnética e conformidade com as restrições de emissão</li></ul> |

Tabela 8 [25-26] – Vantagens e desvantagens da tecnologia PLC

### 3.3.3 RS-485

A norma RS-485 (*Recommended Standard*) é uma norma de comunicação série, definindo as características elétricas dos “drivers” e dos recetores numa rede que utiliza um par de fios para o transporte de dados, e em que a diferença de tensão entre os fios é que define se é um “1” ou um “0” que está a ser transmitido na linha. É uma norma que permite transferências em longas distâncias e em ambientes de ruído. Uma regra que se utiliza para verificar a velocidade de transferência/distância entre dispositivos é que a multiplicação destes fatores não deve ultrapassar  $10^8$ . Assim para um cabo de 50 metros não se deverá ter velocidades superiores a 2 Mbit/s [27]. As topologias de rede que devem ser utilizadas quando se utiliza a norma RS-485 são topologias ponto-a-ponto ou ponto-multiponto (permitindo um máximo de 32 dispositivos conetados) e não topologias em anel ou estrela, devido às reflexões de sinal. A arquitetura que é usualmente utilizada é a arquitetura *master-slave*.

Nestes sistemas é necessário possuir um dispositivo que seja responsável pela transformação dos níveis lógicos, ou seja, fazer a passagem dos níveis lógicos que se verificam no meio (par de fios) para o dispositivo onde serão recebidos os dados (normalmente via micro-controlador) e também no sentido inverso. Estes dispositivos são designados de *tranceivers*. Na figura 29 pode-se observar o esquemático destes dispositivos [28].



**Figura 29 – Esquemático de um *transceiver* RS-485 [28].**

Observando a figura, quando o sinal de controlo relativo à receção /RE for válido é então permitida a receção de dados. Por outro lado se este sinal for inválido verifica-se que tanto as linhas de receção como a de transmissão irão ficar em alta impedância, ou seja, não irão estar ligadas. O mesmo se passa para o sinal de controlo relativo à transmissão DE, ou seja, se este for válido o transmissor está ativo, se for inválido está desativado. A nível do recetor quando o nível elétrico entre as linhas A e B for maior que 200mV então, normalmente, a saída corresponde ao nível lógico positivo; por outro lado quando essa diferença é menor que -200mV a saída corresponde ao nível lógico negativo [28]. Nas tabelas 9 e 10 demonstram-se os níveis lógicos aquando de uma transmissão e de uma receção [29].

De referir que o standard RS-485 apesar de operar apenas com dois fios, aconselha-se a utilização de um terceiro fio que sirva como referência às linhas A e B. Assim consegue-se uma comunicação *half-duplex*, para que se obtenha uma comunicação *full-duplex* é necessária uma configuração utilizando 4 fios, o que não será discutido neste texto. De notar que existem vários protocolos standard que “correm” sobre RS-485 como por exemplo:

- Modbus;
- Profibus;
- HART;
- Sinec H1;
- Interbus.

| Driver |    |    |         |     |
|--------|----|----|---------|-----|
| Inputs |    |    | Outputs |     |
| /RE    | DE | DI | A       | B   |
| X      | 1  | 1  | 1       | 0   |
| X      | 1  | 0  | 0       | 1   |
| 0      | 0  | X  | HZ      | HZ  |
| 1      | 0  | X  | HZ*     | HZ* |

Tabela 9 – Transmissão [29]

\*Transceiver desligado

| Recetor |    |              |         |
|---------|----|--------------|---------|
| Inputs  |    |              | Outputs |
| /RE     | DE | A-B          | R0      |
| 0       | 0  | $\geq +0.2V$ | 1       |
| 0       | 0  | $\leq -0.2V$ | 0       |
| 0       | 0  | Inputs open  | 1       |
| 1       | 0  | X            | HZ*     |

Tabela 10 – Receção [29]

\*Transceiver desligado

Para finalizar esta subsecção é demonstrado na figura 30 um diagrama de blocos que exemplifica uma rede de comunicação utilizando a norma RS-485.

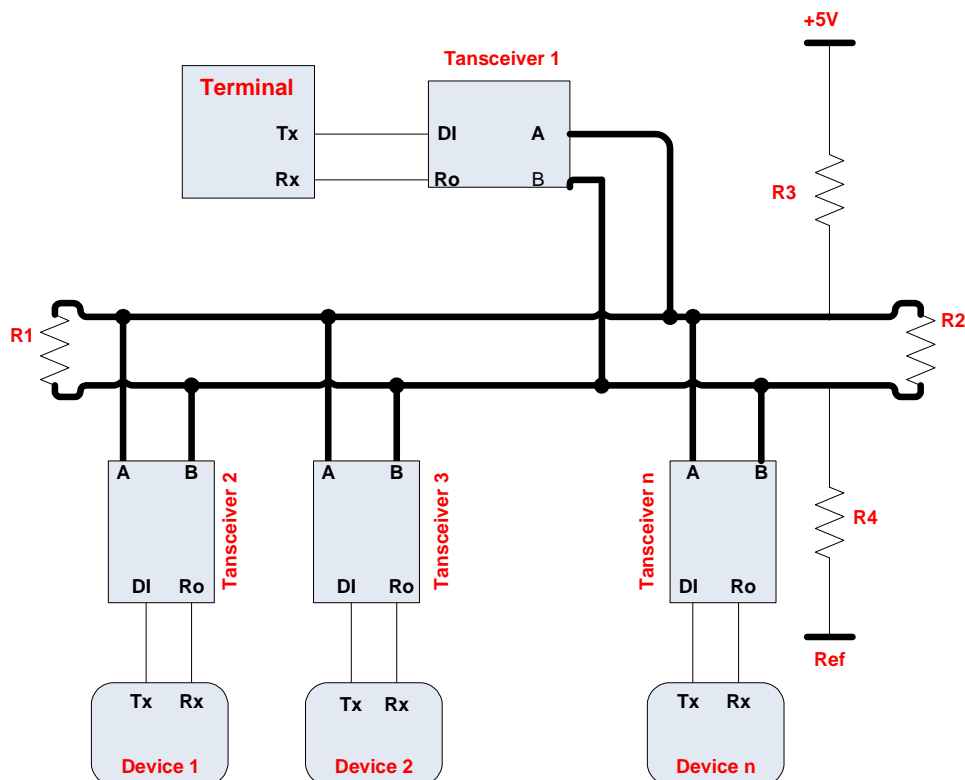


Figura 30 – Diagrama de blocos de um sistema com RS-485.

Apenas de referir que as resistências R1 e R2 são resistências que têm que ser colocadas nas terminações das linhas, com o objetivo de diminuir as reflexões nas linhas, normalmente de valor igual a 120 ohms [30]. As resistências R3 e R4 são resistências de *pull-up* e *pull-down* respetivamente, em que o seu valor se deve situar entre os 450 e os 650 ohms [30]. Os pinos de controlo dos *transceivers* não foram representados no diagrama.

### 3.3.4 RS-232

A norma RS-232 é, à semelhança da RS-485, uma norma de comunicação série que faz, usualmente a conexão entre um DTE (*Data Terminal Equipment*) e um DCE (*Data Circuit Terminating Equipment*). Assim especifica as tensões, temporizações, funções dos sinais, um protocolo para troca de informações e as conexões mecânicas [31].

Os níveis de tensão que correspondem ao nível lógico “1” e ao nível lógico “0” vão de -3 a -15V e de +3 a +15V respetivamente, valores compreendidos entre -3V e +3V não são sinais válidos. Assim verifica-se que os níveis de tensão são superiores aos níveis lógicos que se verifica na maior parte dos circuitos integrados, como tal será também necessário um *transceiver* que faça a devida conversão destes níveis. Para o efeito existem circuitos integrados como o MAX232.

Em relação aos conetores utilizados, o RS-232 especifica 20 sinais diferentes de conexão e os conetores em forma de D são normalmente utilizados, sendo utilizados conetores do tipo macho a nível dos cabos e do tipo fêmea a nível dos dispositivos. Existem também conetores que fazem a conversão de USB para RS-232 para que possa ser feita a ligação série com um PC que não possua porta série.

Em comparação com o *standard* RS-485, o *standard* RS-232 faz uso de três fios, sendo eles o RX, TX e o GND. É, normalmente, utilizado para conexões curtas (quinze metros ou menos). De referir que existem conversores que permitem fazer a conversão de RS-232 para RS-485, o que é útil para se fazer a comunicação com o terminal por exemplo, em sistemas RS-485. É normalmente utilizada comunicação RS-232 para se fazer a comunicação entre o PC e um microcontrolador, como por exemplo o que está representado na figura 6. Para finalizar na figura seguinte demonstra-se um exemplo de comunicação RS-232 entre um terminal e um microcontrolador.

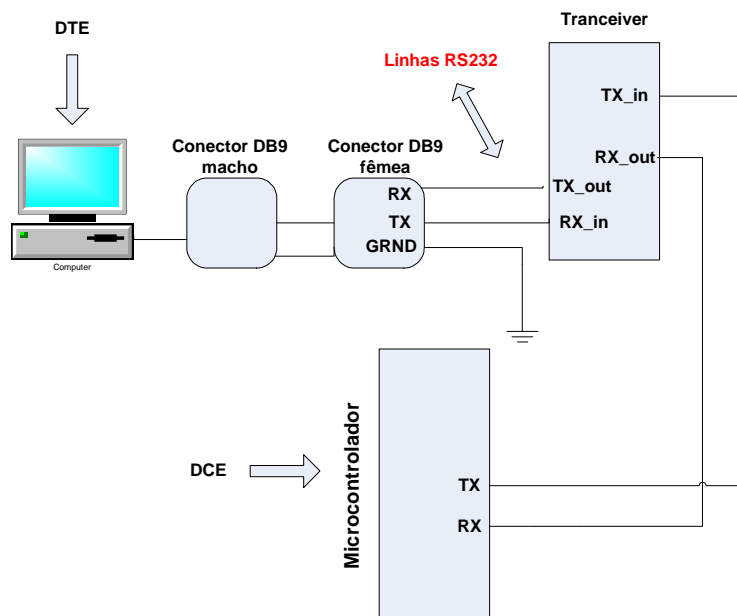


Figura 31 – Exemplo de uma comunicação via RS-232.

### 3.4 Interface de comunicação

Numa rede de comunicações, como por exemplo a descrita na figura 30, verifica-se a existência de *transceivers* para fazer a adaptação dos níveis de tensão do lado do meio físico para o lado do microcontrolador e vice-versa. No entanto, ao nível do controlador terão que existir interfaces de comunicação para que se possa fazer a receção e a transmissão de dados. Nesta secção será feito o estudo do módulo USART que é universal, sendo por isso um dos mais utilizados. De referir que o estudo será orientado ao microcontrolador que está presente na figura 6.

#### 3.4.1 USART

O módulo USART (*Addressable Universal Synchronous Asynchronous Receiver Transmitter*) é um dos três módulos I/O série incorporados em dispositivos PIC18FXX8. Este pode ser configurado como um sistema *full-duplex* assíncrono que pode comunicar com dispositivos periféricos como terminais CRT ou computadores, ou pode ser configurado como um sistema *half-duplex* síncrono que pode comunicar com periféricos como circuitos integrados A/D ou D/A [18]. Assim existem dois registos (de 8 bits) que, são configurados de acordo com a utilização que se quer. Esses registos são:

- **Transmit Status and Control Register (TXSTA);**
- **Receive Status and Control Register (RXSTA);**



Na seguinte figura pode-se observar o registo relativo à transmissão, figura que é complementada com uma tabela que indica qual a função de cada *bit* no registo.

|       |       |       |       |     |       |      |       |
|-------|-------|-------|-------|-----|-------|------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R-1  | R/W-0 |
| CSRC  | TX9   | TXEN  | SYNC  | —   | BRGH  | TRMT | TX9D  |
| bit 7 |       |       |       |     |       |      | bit 0 |

**Figura 32 – Registo TXSTA [18].**

|   |   |
|---|---|
| <b>Clock Source Select bit (CSRC):</b>            | <p>Modo assíncrono: <i>Don't care</i></p> <p>Modo síncrono:</p> <ul style="list-style-type: none"> <li>1 = <i>Master mode</i> (clock gerado internamente).</li> <li>0 = <i>Slave mode</i> (clock gerado externamente).</li> </ul> |
| <b>9-bit transmit enable bit (TX9):</b>           | <p>1 = Selecciona transmissão de 9 bits.</p> <p>0 = Selecciona transmissão de 8 bits.</p>   |
| <b>Transmit Enable bite (TXEN):</b>               | <p>1 = Permite transmissão.</p> <p>0 = Não permite transmissão.</p>   |
| <b>USART Mode Select bit (SYNC):</b>              | <p>1= Modo síncrono.</p> <p>0 = Modo assíncrono.</p>  |
| <b>High Baud Rate Select bit (BRGH):</b>          | <p>Modo síncrono: Não utilizado neste modo.</p> <p>Modo assíncrono:</p> <ul style="list-style-type: none"> <li>1 = Alta velocidade.</li> <li>0 = Baixa velocidade.</li> </ul>   |
| <b>Transmit Shift Register Status bit (TRMT):</b> | <p>1 = Registo TSR vazio.</p> <p>0 = Registo TSR cheio.</p>   |
| <b>9th bit of Transmit Data (TX9D):</b>           | Pode ser um bit de endereçamento/dados ou de um bit de paridade.  |

**Tabela 11 – Descrição dos bits que constituem o registo TXSTA [18]**

De igual modo apresenta-se o registo relativo à receção de dados.

|       |       |       |       |       |      |      |       |
|-------|-------|-------|-------|-------|------|------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0  | R-0  | R-x   |
| SPEN  | RX9   | SREN  | CREN  | ADDEN | FERR | OERR | RX9D  |
| bit 7 |       |       |       |       |      |      | bit 0 |

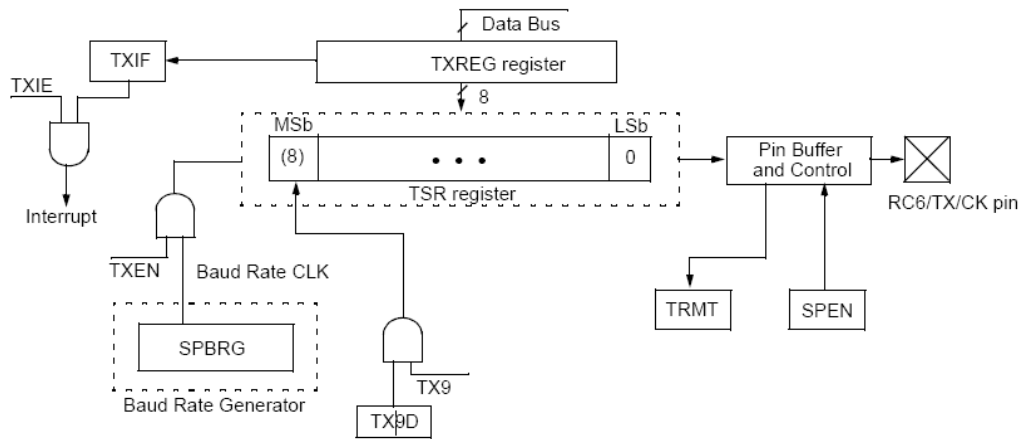
**Figura 33 – Registo RXSTA [18].**

|  |   |
|--|---|
| <i>Serial Port Enable bit (SPEN):</i>        | 1 = Porta série ativada (ativa os pinos TX e RX como portas série).<br>0 = Porta série desativada.  |
| <i>9-bit Receive Enable bit (RX9):</i>       | 1 = Seleciona recepção de 9 bits.<br>0 = Seleciona recepção de 8 bits.  |
| <i>Single Receive Enable bit (SREN):</i>     | Modo assíncrono: <i>Don't care</i> .<br>Modo síncrono – <i>Slave</i> : Não utilizado neste modo.<br>Modo síncrono – <i>Master</i> : <ul style="list-style-type: none"> <li>1 = Ativa recepção simples.</li> <li>Desativa recepção simples.</li> </ul>   |
| <i>Continuous Receive Enable bit (CREN):</i> | Modo síncrono: <ul style="list-style-type: none"> <li>1 = Ativa recepção contínua até que o bit CREN seja colocado a zero (CREN sobrepõe-se a SREN).</li> <li>0 = Desativa recepção contínua.</li> </ul> Modo assíncrono: <ul style="list-style-type: none"> <li>1 = Ativa recepção contínua.</li> <li>0 = Desativa recepção contínua.</li> </ul> |
| <i>Address Detect Enable bit (ADDEN):</i>    | Modo assíncrono (RX9=1): <ul style="list-style-type: none"> <li>1 = Permite a detecção de endereços.</li> <li>0 = Não permite a detecção de endereços.</li> </ul>   |
| <i>Framing Error bit (FERR):</i>             | 1 = Com detecção de <i>Framing error</i> .<br>0 = Sem detecção de <i>Framing error</i> .  |
| <i>Overrun Error bit (OERR):</i>             | 1 = Com detecção de <i>Overrun error</i> .<br>0 = Sem detecção de <i>Overrun error</i> .  |
| <i>9th bit of Received Data (RX9D):</i>      | Pode ser um <i>bit</i> de endereçamento/dados ou um <i>bit</i> de paridade.   |

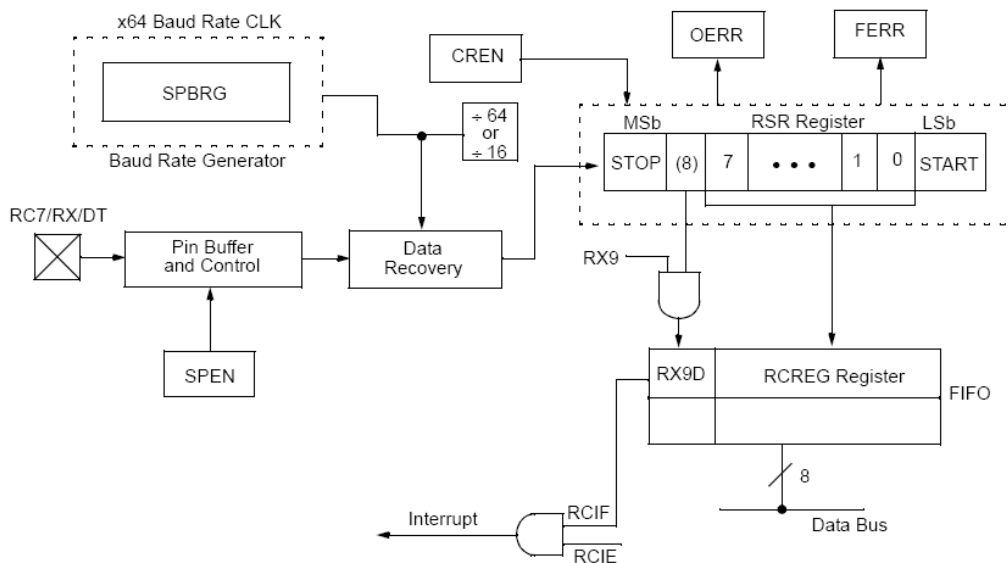
**Tabela 12 – Descrição dos bits que constituem o registo RXSTA [18].**

A amostragem dos dados é feito nos pinos 17 e 18 que correspondem aos pinos TX e RX, respetivamente. De referir que se pode fazer uso das interrupções do microcontrolador gerando uma interrupção sempre que seja feita uma recepção ou uma transmissão. Para isso é necessário colocar a “1” os *bits* TXIE e RXIE. Assim sempre que houver uma interrupção gerada por uma transmissão ou recepção as *flags* TXIF e RXIF irão ser colocadas a “1”.

Para finalizar nas figuras 34 e 35 demonstram-se os diagramas de blocos relativos ao transmissor e ao recetor implementados no microcontrolador PIC [18].



**Figura 34 – Diagrama de blocos relativo ao transmissor [18].**



**Figura 35 - Diagrama de blocos relativo ao recetor [18].**

De referir que estão disponíveis outras interfaces de comunicação série no PIC como o módulo CAN (Controller Area Network) e módulo MSSP (Master Synchronous Serial Port Module) que permitem fazer a comunicação com outros microcontroladores ou periféricos. No entanto estes módulos são de utilização mais complexa, não se justificando a sua utilização no tipo de aplicação em causa.

### 3.5 Em resumo

Como já foi dito a integração de comunicação em sistemas de iluminação pública é de extrema importância. Por conseguinte, o objetivo deste capítulo foi fazer um estudo acerca de redes de comunicações. Assim, começou-se por apresentar e explicar as diversas topologias de rede que podem ser adotadas:

- ✓ **Topologia de rede em barramento:** Topologia em que todos os nós estão ligados ao mesmo barramento físico de dados, sendo assim o barramento diz-se partilhado.
- ✓ **Topologia de rede ponto-a-ponto:** Topologia em que é criada uma ligação direta entre os nós da rede.
- ✓ **Topologia de rede em estrela:** Utiliza um *hub* central em que as ligações com os restantes nós da rede são ponto-a-ponto.
- ✓ **Topologia de rede em anel:** Os nós são distribuídos em anel, onde os dados circulam em torno deste numa dada direção.
- ✓ **Topologia de rede em árvore:** É uma rede hierárquica, em que existe um nó central que está no topo da rede, ligando-se a um ou mais nós que irão fazer parte do segundo nível da rede e assim sucessivamente.
- ✓ **Topologia de rede em malha:** Topologia em que todos os nós estão interligados, fazendo com que a rede seja flexível e redundante.

Seguidamente fez-se um estudo acerca de como é controlada a troca de dados, entre os diferentes elementos da rede. Apresentaram-se as seguintes arquiteturas de rede:

- **Arquitetura master-slave:** Existe um dispositivo que é considerado como *master*, sendo este que controla todas as trocas de dados que existam na rede.
- **Arquitetura master-slave dinâmica:** Esquema em que, o primeiro nó a iniciar uma transferência de dados é considerado como *master*.
- **Arquitetura multimaster:** Como o nome indica, todos os dispositivos são considerados como *master*.
- **Arquitetura peer-to-peer:** Todos os dispositivos da rede desempenham o mesmo papel.

- **Arquitetura centralizada:** Existe um dispositivo central que é responsável pela gestão da rede, podendo os outros dispositivos efetuar trocas entre si sem ter que passar pelo dispositivo central.

Foram também estudadas quatro formas de se fazer a transferência de dados, sendo elas:

- ❖ **Wireless:** Tecnologia em que a transferência de dados não necessita de qualquer cabo, sendo bastante importante a banda de frequência que é utilizada.
- ❖ **Power line communication:** Tecnologia que faz uso da rede de distribuição elétrica para fazer a transferência de dados. Assim no mesmo cabo onde é feita a alimentação do sistema, também irão circular dados de controlo deste.
- ❖ **RS-485:** Standard de comunicação série que faz uso de um par de fios para o transporte de dados, em que a diferença de tensão entre os fios é que define se é um “1” ou um “0” que está a ser transmitido na linha. Suporta transferência em grandes distâncias.
- ❖ **RS-232:** Também um standard de comunicação série que faz uso de três fios para o transporte de dados, que é, geralmente, feito entre um DTE e um DCE. Não suporta transferência em grandes distâncias.

Por fim foi feito um estudo acerca de um dos interfaces de comunicação série presente no microcontrolador (PIC18FXX8), sendo ele o módulo USART. Este possui dois registos de estado e controlo, que devidamente configurados possibilitam um dos seguintes modos de funcionamento:

- Modo assíncrono *full-duplex*;
- Modo síncrono *half-duplex – Master*;
- Modo síncrono *half-duplex – Slave*;



# **Capítulo IV**

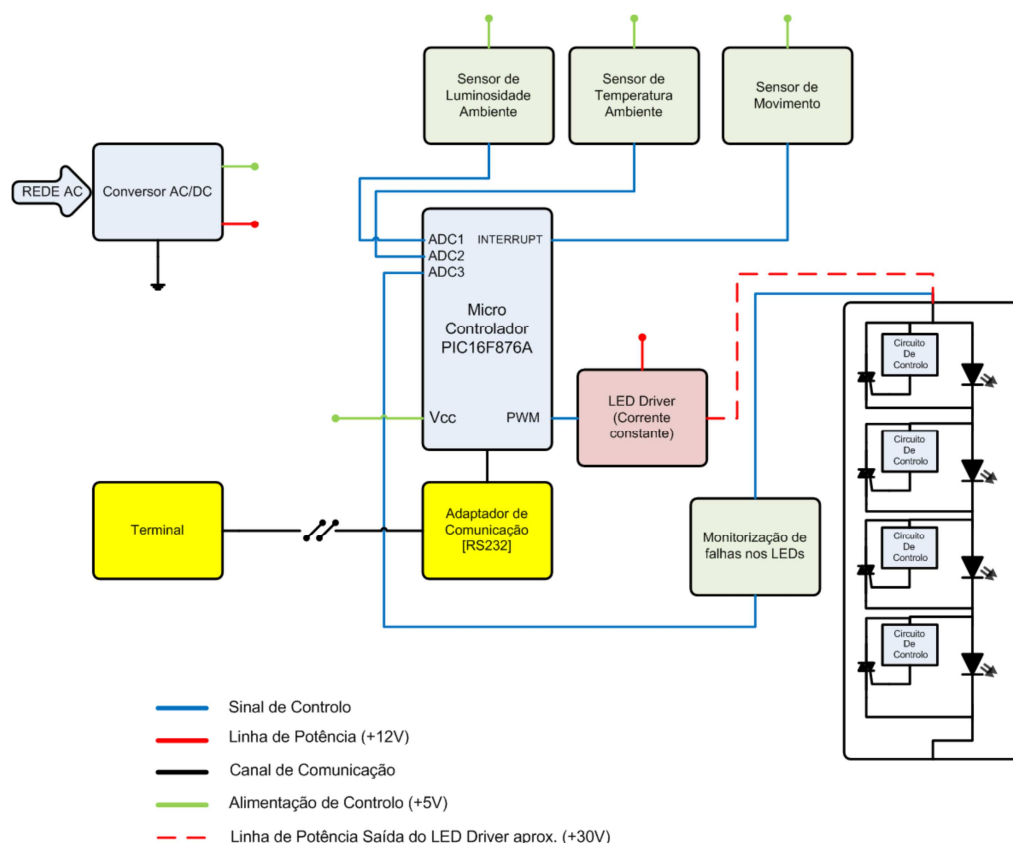
## **4 Sistema final**

Após todas as considerações feitas nos capítulos anteriores acerca de comunicações e variáveis de medida, tendo em conta os sistemas de iluminação pública, foram desenvolvidos dois protótipos de teste. De referir que ambos os protótipos assentam num dispositivo anteriormente desenvolvido. Os principais objetivos deste capítulo são: desenvolver um detetor de falhas para os LEDs, implementar um sensor de *tilt* que permita aferir sobre a inclinação da estrutura de iluminação e por último desenvolver uma rede de comunicações entre os dois protótipos e um terminal que tenha controlo sobre o sistema.

### 4.1 Protótipo de referência

Como já foi dito, neste projeto pretende-se fazer evoluir um protótipo já desenvolvido, também no âmbito de uma tese de mestrado inserida no projeto LITES. Assim, é de todo o interesse fazer uma breve introdução das características do protótipo utilizado e quais os seus principais parâmetros. Por conseguinte, é objetivo desta secção fazer um breve estudo do sistema já implementado e delinear quais os objetivos para o seu desenvolvimento.

No capítulo II foram introduzidas várias topologias de controlo que podem ser empregues na implementação de sistemas de iluminação. O protótipo utilizado insere-se na topologia de um sistema autónomo com controlo ativo de energia. Na figura 36 [32] apresenta-se um diagrama de blocos do sistema já desenvolvido.



**Figura 36 – Arquitetura implementada [32].**

Como se pode observar na figura a unidade de controlo escolhida foi o microcontrolador PIC16F876A. Agora será feita uma breve descrição dos principais blocos do sistema.



A alimentação da luminária é feita por um módulo de conversão AC/DC de comutação, uma vez que a distribuição de energia da rede pública é feita em AC e os LEDs são dispositivos que operam em DC. A tensão de saída do conversor é de 12V e a potência fornecida é de 24W.

A nível sensorial, este protótipo já possui um conjunto de três sensores que são responsáveis pela medida de três variáveis importantes: o sistema está equipado com um sensor de movimento, um sensor de luminosidade ambiente e um sensor de temperatura. O sensor de movimento utilizado foi do tipo IVP (infravermelho passivo), o sensor de luminosidade baseia-se num fotodíodo (TEPT – 5700) e por último o sensor de temperatura consiste num sensor integrado, o LM335.

O módulo de alimentação dos LEDs passa por uma fonte de corrente constante, com controlo pulsado. Estes são alimentados a partir de reguladores de comutação DC/DC. O conversor utilizado foi do tipo *boost*, uma vez que o arranjo de LEDs é constituído por 10 LEDs de aproximadamente 3.2V, sendo necessário um conversor que para uma entrada de 12V coloque à saída 32V. O LED-Driver (LD) utilizado foi o LM3424 que possui um pino externo onde se pode proceder à variação do fluxo luminoso, a partir de um controlo pulsado (PWM).

Por último, este sistema já tem algum mecanismo de monitorização de falhas nos LEDs. No entanto, apresenta algumas limitações que irão ser discutidas na secção seguinte, bem como as opções tomadas para contornar essas mesmas limitações. De referir que nos terminais de cada LED existe um circuito constituído por um Tiristor e um circuito de controlo que permite uma alternativa à passagem de corrente, no caso em que um ou mais LEDs falhem, garantindo assim a passagem de corrente e que a falha de um LED não signifique a falha de todos (uma vez que estão dispostos em série).

Uma vez apresentadas as principais características e funcionalidades do protótipo utilizado, será então descrito, nas secções seguintes, o desenvolvimento efetuado no âmbito desta tese de mestrado. Este desenvolvimento passa por três novas funcionalidades.

- Detecção otimizada de falhas nos LEDs;
- Inclusão de um sensor de integridade do sistema;
- Desenvolvimento de uma rede de comunicação entre dois protótipos e um terminal.

### 4.2 Detetor de falhas nos LEDs

Como já foi dito pretende-se que o sistema desenvolvido seja capaz de detetar, caso ocorram, falhas que possam existir em um ou mais LEDs. Como o sistema desenvolvido consiste numa luminária que possui um arranjo com 10 LEDs, até quatro falhas é suportável, a partir do momento em que ocorram 5 falhas já se tem o sistema a funcionar a 50% e, como tal, será necessário prestar a devida reparação. O arranjo de LEDs está de acordo com a figura 36. Como se pode observar esta configuração é robusta no que diz respeito a falhas, permitindo uma alternativa à passagem de corrente aquando da falha de um LED, fazendo com que uma falha não implique a falha de todos os LEDs.

No protótipo desenvolvido já é feita uma iniciação à monitorização de falhas nos LEDs. No entanto, a solução desenvolvida foi pouco otimizada e por conseguinte, apresenta limitações. Assim se para uma falha num LED o sistema detetava devidamente, para duas falhas já ocorriam alguns problemas. É necessário ter em conta alguns aspetos pertinentes no desenvolvimento do detetor, nomeadamente no que diz respeito às limitações da tensão de saída no próprio conversor, bem como a tensão de rutura do condensador que é colocado à saída (apresenta uma tensão de rutura de 50V). No entanto o grande problema do sistema implementado encontra-se nos dispositivos de proteção de falhas nos LEDs. Na figura 37 está representada a característica I-V destes dispositivos.

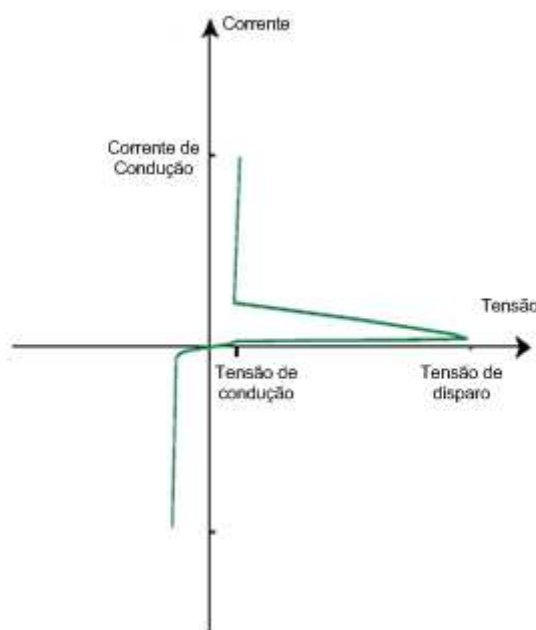


Figura 37 – Característica I-V dos Tiristor.

Como se pode observar na figura, quando estes dispositivos atingem uma certa tensão começam a conduzir, diminuindo a tensão aos seus terminais. O problema encontra-se na tensão de disparo destes componentes. Por conseguinte, no protótipo, esta tensão encontra-se compreendida entre os 9 e os 18V, que são tensões relativamente altas e que limitam o número de falhas que se pode detetar. O que acontecia para duas falhas é que no instante de disparo dos dispositivos, a tensão de saída era relativamente elevada e verificava-se que os protetores não conseguiam baixar a sua tensão para a tensão de condução, ficando então a dissipar energia e aquecendo demasiado. Neste ponto, o sistema já não se comporta como o esperado e será necessário fazer a devida notificação. Assim é necessário recorrer a dispositivos que apresentem uma tensão de disparo mais baixa de modo a poderem ser feitas mais deteções. Para isso o dispositivo selecionado foi o PLED5HT SOT 89 Series da Littlefuse, que apresenta tensões de disparo da ordem dos 5V e tensões de condução na ordem dos 1.6V para uma corrente de 700mA [33].

O projecto do detetor de falhas passa pelos seguintes passos:

1. Definir uma tensão normal de funcionamento, em que todos os LEDs estão ativos.
2. Definir uma tensão máxima que corresponde ao mau funcionamento de um dos protetores e como tal o sistema já não funciona como o esperado, logo será necessário desligar o LD.
3. Definir um conjunto de 5 níveis abaixo da tensão normal de funcionamento e que correspondem à deteção de falhas nos LEDs.

Na figura 38 pode-se observar um diagrama que exemplifica o que foi descrito acima. De notar que para além da deteção de falhas em 5 LEDs já não valerá a pena detetar mais falhas, uma vez que o sistema já estará a funcionar a 50 %.

A forma de se chegar aos respetivos níveis é bastante simples e é de acordo com as seguintes expressões:

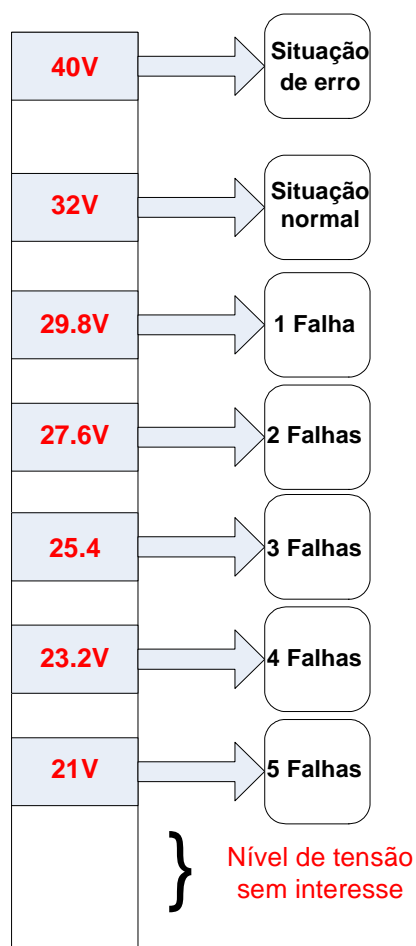
$$v=3.2V \times (\text{nr de LEDs}-Y)+1V \times Y \quad (1)$$

em que, nr de LEDs=10 e Y corresponde ao número de falhas detetadas. Como a corrente que atravessa o arranjo de LEDs é de 350mA, considera-se que a tensão de condução dos PLEDs está limitada entre os valores presentes nas expressões (1) e (2). Na figura 38 é

considerada a situação limite em que os LEDs estão a funcionar a 3.2V (que é valor fornecido pelo fabricante [34]) e o circuito protetor apresenta 1V (1) quando a conduzir. A nível de eletrónica existem sempre certas variações de tensão e, como tal, é necessário ter em conta essas flutuações.

$$v = 3V \times (\text{nr de LEDs} - Y) + 0.8V \times Y \quad (2)$$

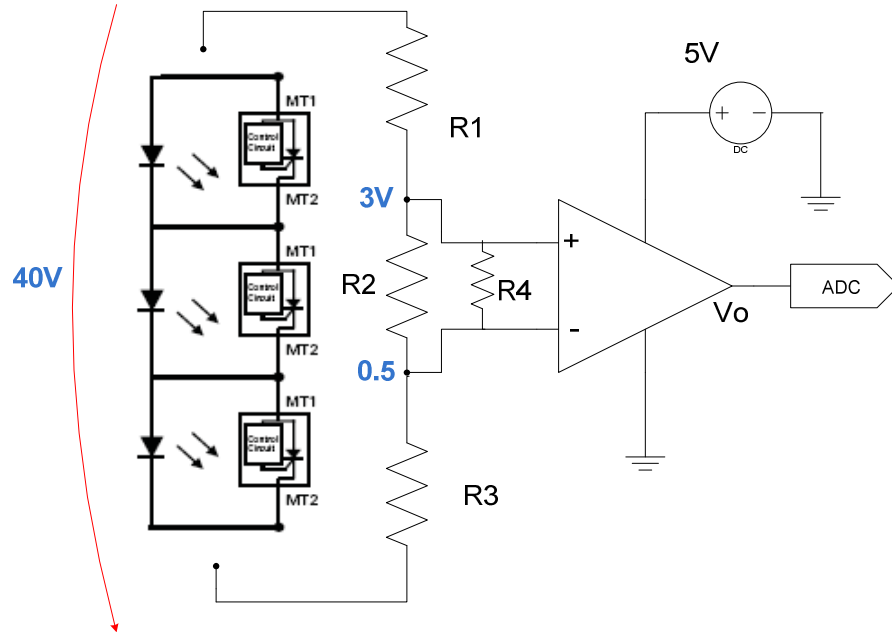
Assim, na expressão (2) calcula-se os níveis tendo em conta 3V para a tensão dos LEDs e 0.8V para os PLEDs. É criado um intervalo de valores para cada uma das ocorrências possíveis, ou seja, 1, 2, 3, 4 ou 5 falhas detetadas, minimizando pois a ocorrência de erros de leitura.



**Figura 38 – Diagrama correspondente à implementação do detetor de falhas.**

Uma vez calculados os diferentes níveis é então necessário introduzir alguma eletrônica para colocar os níveis de tensão numa gama que é suportável pelo

microcontrolador utilizado ([0...5V]). Na figura 39 está demonstrado o circuito básico responsável pela conversão entre gamas.



**Figura 39 – Circuito básico de conversão de gamas.**

Olhando para o circuito da figura 39 pode-se verificar que este é projetado para a situação limite em que aparecem 40V aos terminais da associação de LEDs (o que corresponde à situação de erro descrita na figura 38), aparecendo assim aos terminais da resistência R2 2.5V, correspondendo à situação limite em que a entrada da ADC é de 5V, uma vez que o amplificador apresenta um ganho  $A_v=2$ . O amplificador de instrumentação utilizado foi AD623 e, para que este apresente um ganho de 2 a resistência R4 terá que ser de 100K, o que pode ser verificado na informação do componente [35]. A necessidade de se inserir ganho no circuito advém do facto de o amplificador não ser *rail-to-rail* à entrada. A tensão à entrada não inversora deste não pode exceder os 3.5V [35]. O cálculo das resistências é relativamente simples e baseia-se nas duas equações seguintes:

$$2.5 = \frac{R_2}{R_2 + R_1 + R_3} \times 40 \quad (3)$$

$$0.5 = \frac{R_3}{R_3 + R_2 + R_1} \times 40 \quad (4)$$

Desenvolvendo as expressões (3) e (4) chega-se à seguinte relação entre as resistências:

$$R3=0.0135 \times R1 \quad (5)$$

$$R2= 5.07 \times R3 \quad (6)$$

Sendo escolhido um valor pertinente para  $R1$  é então possível calcular-se as resistências  $R3$  e  $R2$  utilizando as expressões (5) e (6). De notar que  $R1$  deve ser a resistência de maior valor. Assim para  $R1=680k$ , obtém-se  $R3=10k$  e  $R2=47k$ .

Na prática, os valores medidos para os intervalos que vão delimitar as falhas ocorridas são relativamente diferentes dos valores teóricos obtidos nas equações (1) e (2), uma vez que a tensão aos terminais dos LEDs era ligeiramente diferente, na ordem dos 2.9V. Em relação aos protetores, estes apresentaram uma tensão de condução de 0.9V. No gráfico da figura 40 é então possível observar os níveis obtidos para o detetor desenvolvido.

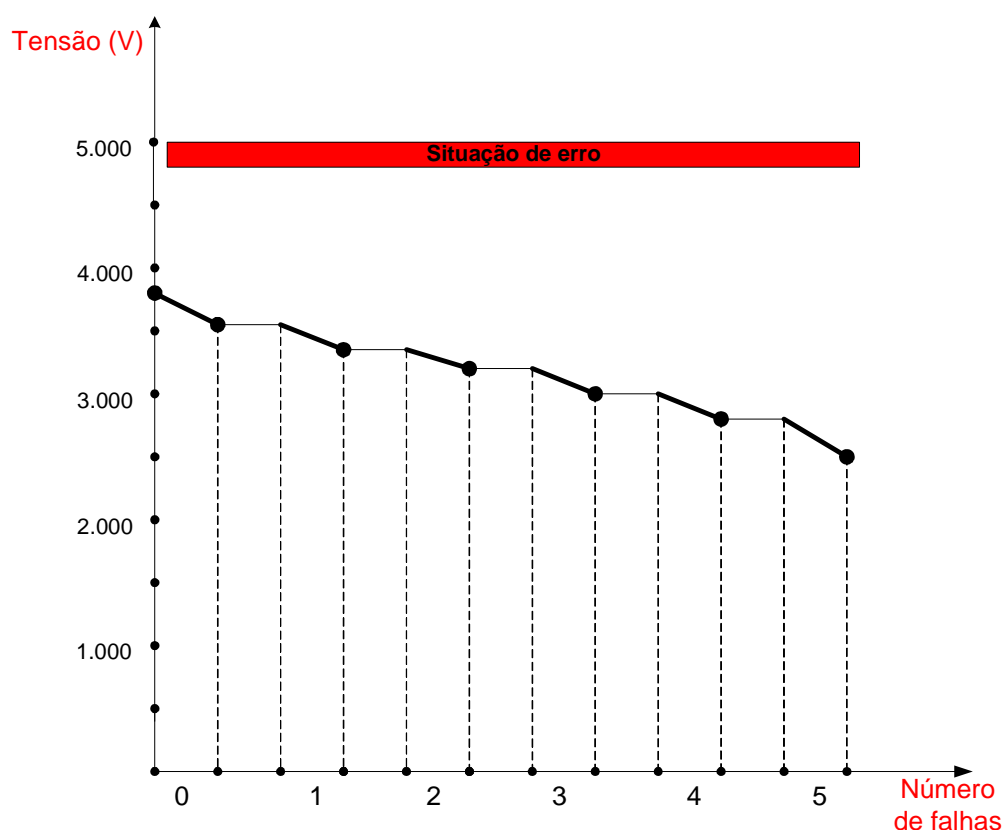


Figura 40 – Calibração do detetor de falhas.

Para complementar o gráfico pode ser observada a seguinte tabela, demonstrando exatamente quais os níveis referentes a cada falha.

| Número de falhas | Intervalo em tensão (V)                   |
|------------------|---|
| Situação de erro | $4.800 \leq \text{Valor lido} \leq 5.000$ |
| Sem falhas       | $3.570 \leq \text{Valor lido} \leq 3.800$ |
| 1                | $3.385 \leq \text{Valor lido} < 3.570$    |
| 2                | $3.200 \leq \text{Valor lido} < 3.385$    |
| 3                | $3.000 \leq \text{Valor lido} < 3.200$    |
| 4                | $2.800 \leq \text{Valor lido} < 3.000$    |
| 5                | $2.500 \leq \text{Valor lido} < 2.800$    |

Tabela 13 – Níveis obtidos para o detetor

De notar que algumas dificuldades foram encontradas na calibração do detetor, nomeadamente a interferência do aquecimento dos LEDs nas leituras. Por conseguinte foi necessária a inclusão de um dissipador atrás da placa de LEDs. O outro principal entrave foi relativo à zona onde era feita a leitura, ou seja, como a corrente nos LEDs é pulsada, (o que varia é o tempo em que eles estão em condução) é pertinente saber onde é feita a leitura. A solução mais simples foi colocar o *duty-cycle* a 100% antes de ser feita uma leitura, o que constitui uma limitação do detetor. As leituras são feitas sempre que se detete movimento uma vez que, o fluxo é colocado ao máximo.

### 4.3 Sensor de Tilt

Uma importante variável no âmbito de iluminação pública é, como já foi dito, a integridade do sistema, como se verifica na figura 16. Assim, neste caso, pretende-se que o sistema seja capaz de detetar, caso ocorra, uma certa inclinação na estrutura de iluminação. Ou seja, a partir de uma certa inclinação o sistema deve ser imediatamente desligado. Neste âmbito foram utilizados os chamados sensores de *tilt* que permitem fazer exatamente o que foi descrito anteriormente.

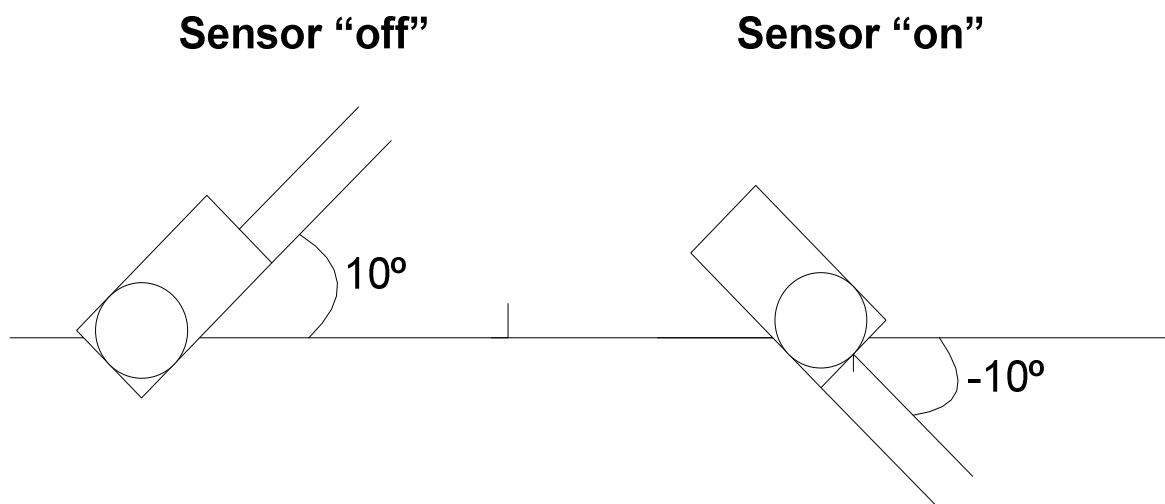
Antes de se mostrar qual a configuração utilizada para se detetar a inclinação é pertinente explicar o funcionamento do sensor utilizado. Assim o sensor utilizado foi o cw1300-1 da Comus Interntional, que é um sensor não muito preciso mas que, para o que se pretende, serve perfeitamente, tendo também a vantagem de ser bastante barato. O funcionamento deste sensor assemelha-se ao funcionamento de um interruptor, ou seja, para uma certa inclinação o sensor comporta-se como um circuito aberto, estando “off” e

para outro valor este comporta-se como um curto-circuito, estando “on”. Na figura 41 pode-se observar o sensor utilizado [36].



**Figura 41 - Sensor de *tilt* utilizado [36].**

Como se pode observar na figura, este sensor consiste em dois contactos metálicos e um cilindro que possui uma esfera no seu interior. Da informação do fabricante do componente pode-se verificar que, para uma inclinação superior a 10 graus, em relação à horizontal, o sensor está “off”. Por outro lado, se este apresentar uma inclinação inferior a menos 10 graus, está “on” [36]. O que acontece no primeiro caso é que a esfera que está no interior do cilindro irá estar encostada à parte detrás do sensor, mantendo assim os dois contactos em aberto. No segundo caso a esfera irá estar encostada aos dois contactos fechando assim o circuito. Este comportamento pode ser observado na figura seguinte.

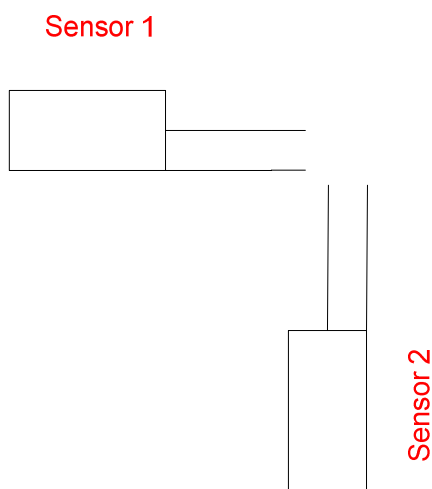


**Figura 42 – Funcionamento do sensor de *tilt*.**

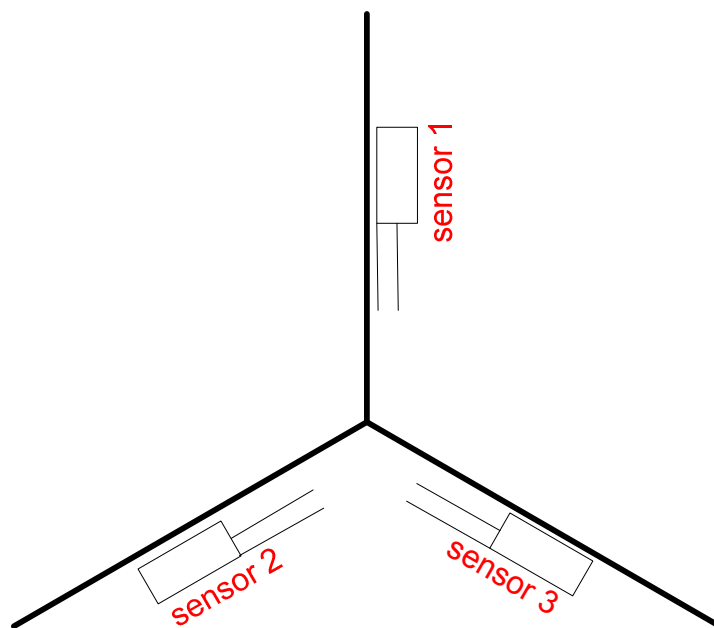
Uma vez descrito o funcionamento do sensor é então possível apresentar a configuração adotada para se fazer uma deteção omnidirecional da inclinação de uma estrutura de iluminação. Como é facilmente perceptível pelo funcionamento do sensor não é possível



fazer a detecção em todas as direções utilizando apenas um sensor. Com uma configuração utilizando dois sensores colocados na horizontal e formando um ângulo de  $90^\circ$ , como demonstrado na figura 43 também não se consegue fazer uma detecção em todas as direções, uma vez que vai existir uma zona na qual nenhum dos dois sensores irá disparar. Como tal foi necessário recorrer a uma configuração utilizando três sensores na horizontal, formando um ângulo de  $120$  graus entre si, como se pode observar na figura 44. Assim consegue-se que haja disparo em todas as direções. De notar que nas direções entre cada dois sensores o disparo não é equivalente às direções onde estes se encontram, uma vez que nestes casos há que ter em conta a decomposição do vetor (sentido), sendo necessário fazer um estudo trigonométrico para aferir qual o ângulo de disparo, o qual não é feito neste texto. A direção de mais difícil detecção será aquela coincidente com o ponto intermédio entre dois sensores, o que corresponde a um ângulo de ativação maior. De referir ainda que a construção desta configuração deve ser o mais simétrica possível, estando os sensores à mesma distância da origem e possuindo a mesma inclinação inicial, inclinação esta que tem que garantir que estes se encontrem no estado “off”. De notar que na figura não esta representada a inclinação inicial.



**Figura 43 – Configuração utilizando dois sensores.**



**Figura 44 – Configuração utilizando 3 sensores.**

Uma vez explicada a implementação do sensor de inclinação é então pertinente explicar como foram feitas as medições dos ângulos. O teste efetuado consistiu num varrimento de 360 graus com uma resolução de 20 graus e medir o ângulo para o qual um dos sensores disparava. Para isso assentou-se a configuração de sensores numa plataforma circular e, por conseguinte para cada ponto de medição, fixa-se o ponto e eleva-se a plataforma pelo ponto diametralmente oposto e medindo-se a altura para a qual um dos sensores disparava. Finalmente e por simples utilização do teorema de Pitágoras torna-se óbvio o cálculo do ângulo correspondente, em que a altura é medida e a hipotenusa corresponde ao diâmetro da circunferência (13.5 cm). Na figura 45 está ilustrada a construção utilizada para se fazerem as medidas descritas anteriormente.

Dito isto, são então apresentados os resultados obtidos na figura 46, complementados pela tabela 14.

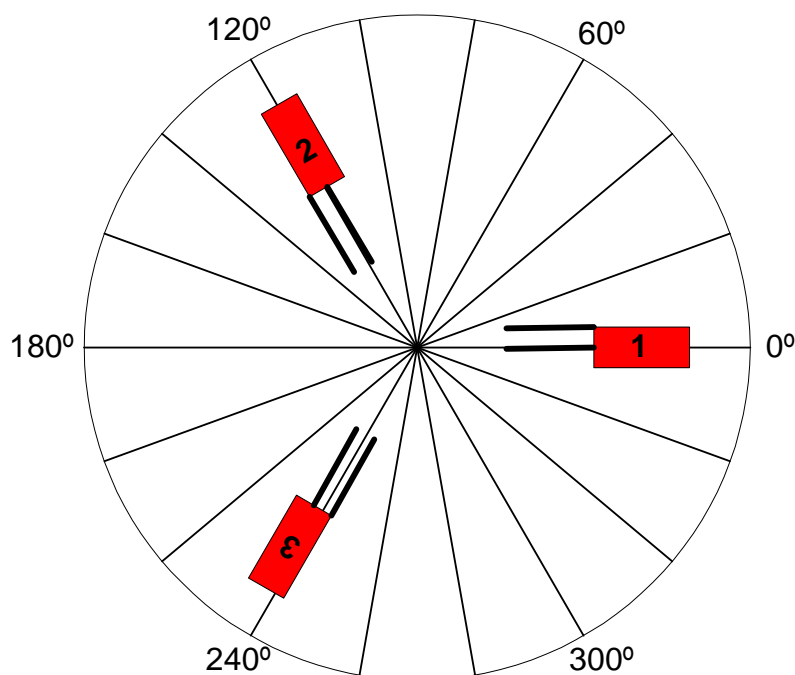
Por observação do gráfico verificámos um comportamento esperado, tendo em conta que os eixos onde se encontram os três sensores correspondem a  $0^\circ$ ,  $120^\circ$  e  $240^\circ$ .

Assim para o ponto  $0^\circ$  eleva-se a plataforma pelo ponto diametralmente oposto e como se pode observar na figura 45, será o sensor 2 ou o sensor 3 o responsável pela ativação. Esta situação corresponde ao caso em que se verifica um ponto intermédio entre

dois sensores, obtendo assim um ângulo máximo. O mesmo acontece fixando o ponto  $120^\circ$  e elevando pelo ponto  $300^\circ$  e fixando o ponto  $240^\circ$  elevando pelo ponto  $60^\circ$ . Obtém-se assim três máximos. No sentido contrário, ou seja, fixando os pontos  $180^\circ$ ,  $300^\circ$  e  $60^\circ$  e elevando pelos pontos  $0^\circ$ ,  $120^\circ$  e  $240^\circ$  respetivamente, medem-se pontos de mínimo uma vez que a ativação é influenciada apenas por um sensor.

De notar que no ponto referente a  $240^\circ$  o valor medido foi relativamente maior uma vez que a construção do protótipo bem como a medição dos valores foi manual e sendo assim existem sempre erros de medição. Por outro lado como a construção foi manual o protótipo não é exatamente simétrico. Por fim, é de referir que os ângulos medidos são relativamente altos, uma vez que a inclinação inicial dos sensores foi ligeiramente alta, no entanto pode-se concluir que o sensor implementado é funcional.

Uma vez compreendido como fazer a deteção omnidirecional, e tendo medido os ângulos de ativação, o princípio de funcionamento para interligar os sensores com a unidade de controlo é bastante simples. Assim os três sensores vão estar na configuração descrita atrás, ligados em paralelo, para que quando um detete inclinação seja ativada uma interrupção externa, desligando-se assim a estrutura de iluminação correspondente. Na figura 47 é mostrado o princípio base para a interligação com a unidade de controlo.



**Figura 45 – Esquemático de como foram obtidos os resultados.**

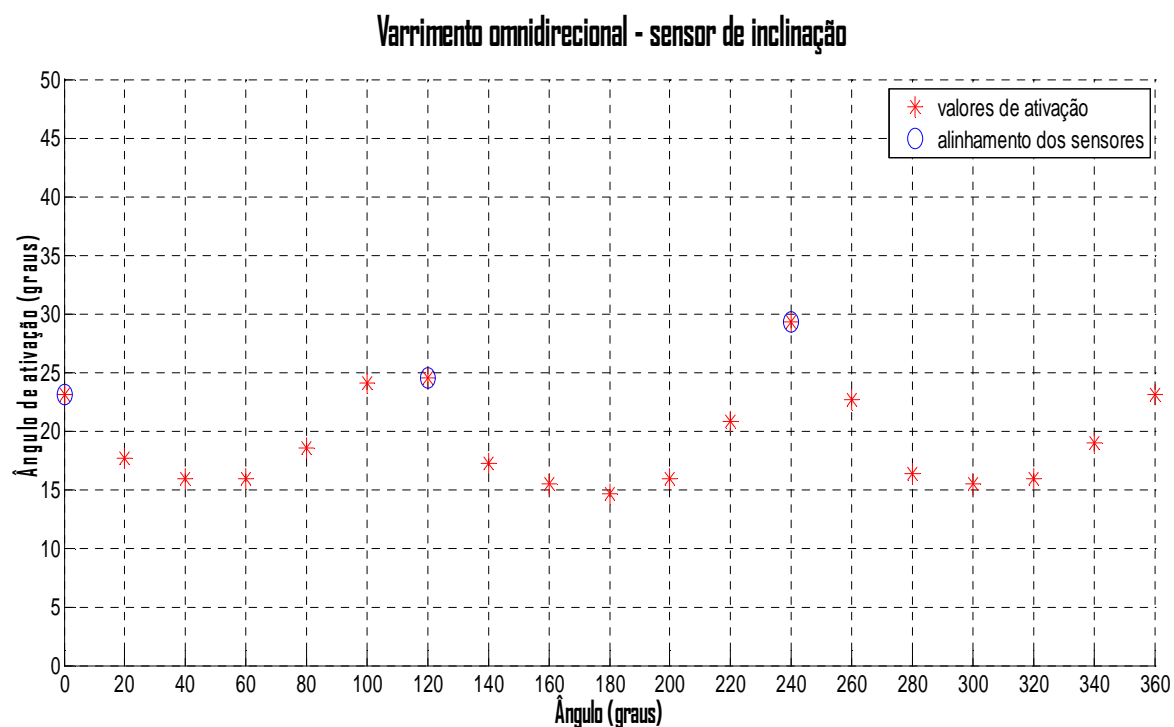
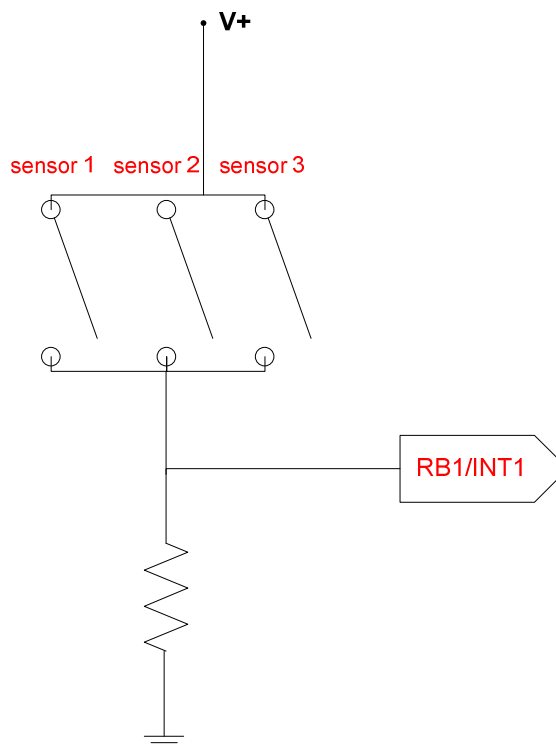


Figura 46 – Gráfico com os resultados obtidos.

| $\theta$ (graus) | $\alpha[\theta]$ (graus) |
|------------------|--------------------------|
| 0                | 23.12                    |
| 20               | 17.68                    |
| 40               | 15.91                    |
| 60               | 15.91                    |
| 80               | 18.57                    |
| 100              | 24.04                    |
| 120              | 24.51                    |
| 140              | 17.24                    |
| 160              | 15.47                    |
| 180              | 14.59                    |
| 200              | 15.91                    |
| 220              | 20.83                    |
| 240              | 29.27                    |
| 260              | 22.66                    |
| 280              | 16.35                    |
| 300              | 15.47                    |
| 320              | 15.96                    |
| 340              | 19.02                    |

Tabela 14 – Resultados obtidos para o sensor de inclinação



**Figura 47 – Esquemático da interligação com a unidade de controlo.**

Embora o princípio de funcionamento seja simples, a nível de software o caso não é tão trivial, pois é preciso ter em atenção alguns aspetos climáticos que podem influenciar o seu desempenho. Assim, o sistema tem de suportar três situações:

1. Poste inclinado com retorno à posição original;
2. Poste inclinado sem retorno à posição original;
3. Estado intermitente em que o poste acaba por inclinar, ou não.

No que diz respeito ao primeiro caso pretende-se que a deteção não ocorra uma vez que o poste voltou à posição original, o que já não se verifica no segundo caso. O terceiro caso diz respeito, por exemplo, a um dia em que o vento seja forte, em que o poste esteja a abanar repetidamente até que acabe por inclinar ou voltar à posição inicial, sendo por isso necessário atuar de acordo com a posição final. De um modo muito simples é apresentado um fluxograma que explica a implementação do *software* que permite fazer as deteções atrás descritas. Na figura 48 está representado o fluxograma relativo ao *software* implementado para o sensor de *tilt*. Seguidamente enunciam-se alguns aspetos fundamentais para a compreensão do fluxograma.

- As variáveis *t0Count* e *Tilt\_off* são contadores a ser inicializadas na rotina de atendimento à interrupção externa.
- *Tilt\_off* é um contador que indica o número de “0’s” relativos ao sensor, em que este não apresenta uma inclinação suficiente para ativar a interrupção.
- A variável *t0Count* garante que as leituras (na rotina de atendimento à interrupção do *timer0*) sejam feitas durante três segundos (*t0Count*=229).

Desta forma garante-se que sempre que se entre na rotina de atendimento ao *timer*, os contadores estejam devidamente inicializados, permitindo assim fazer uma detecção correta.

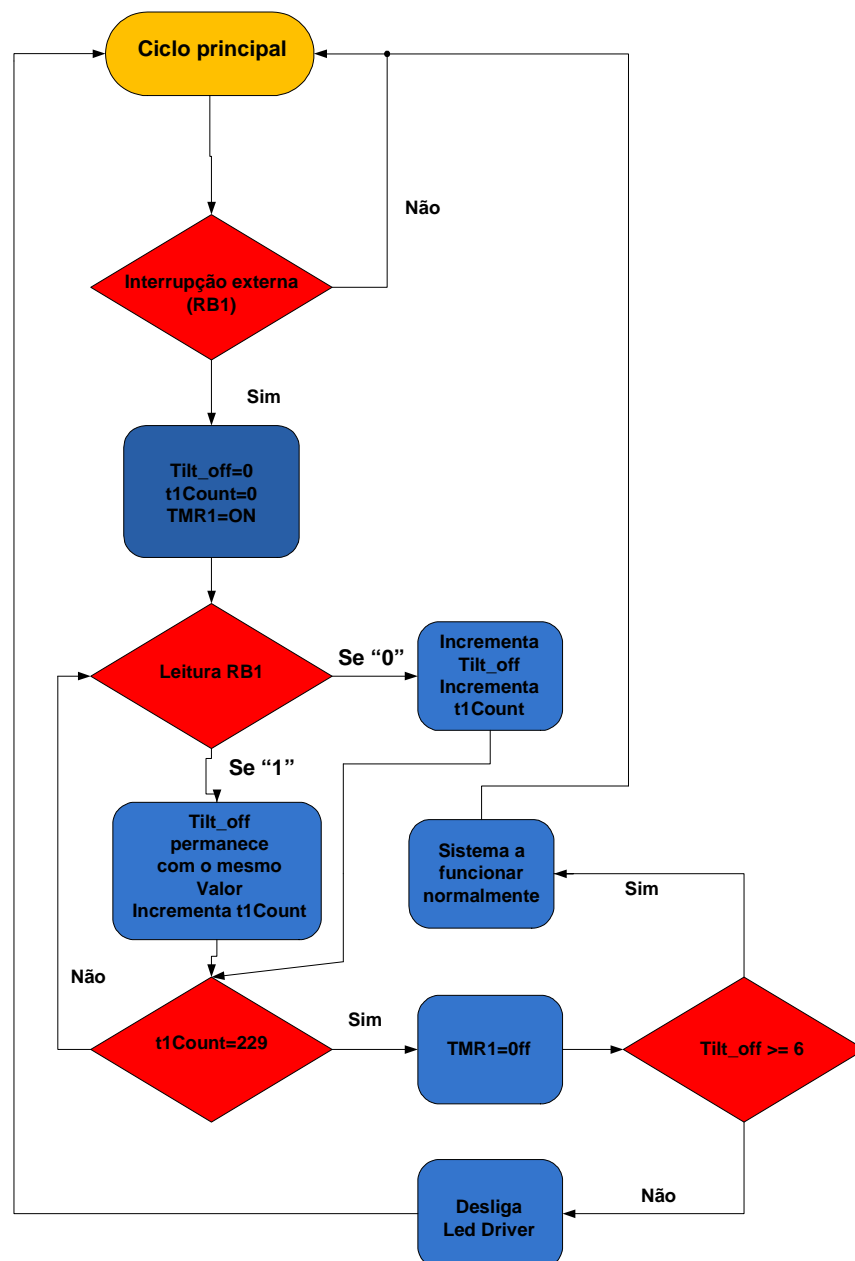


Figura 48 – Fluxograma do software do sensor de inclinação.

## 4.4 Integração de Comunicação

Uma vez discutidas as implementações anteriores, serão então discutidas quais as opções e estratégias tomadas com vista à integração de comunicação, sendo este o principal objetivo desta tese de mestrado, uma vez que introduz uma inovação relativamente ao protótipo apresentado na figura 36.

Esta secção irá retratar o *hardware* implementado, bem como o *software* necessário para que não ocorram falhas nas trocas de dados entre os dispositivos. De referir que também será feita uma breve abordagem teórica ao protocolo de comunicação que é utilizado.

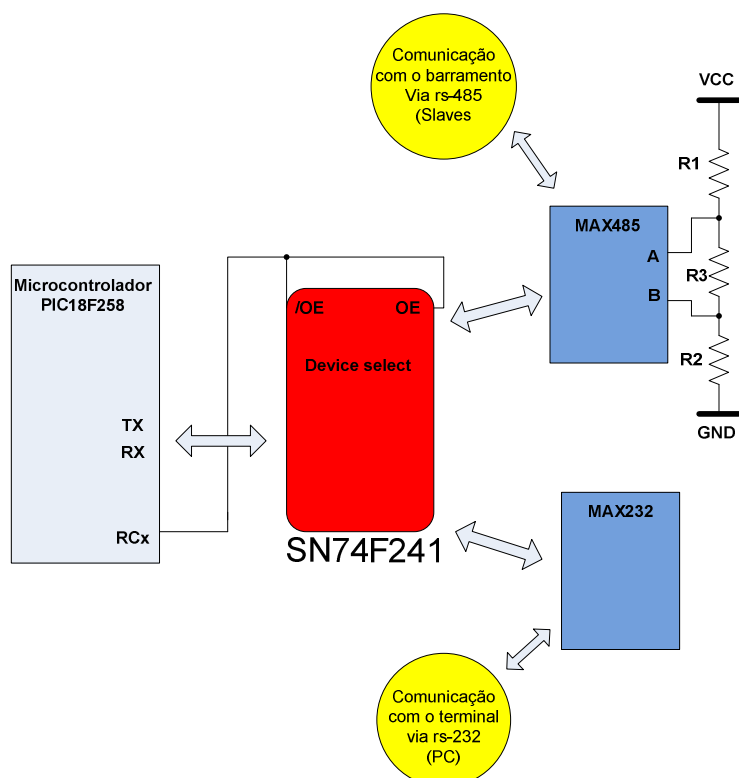
### 4.4.1 Hardware implementado

No capítulo III foram feitas várias referências a diferentes topologias de rede, a diferentes arquiteturas de rede e a diferentes meios onde os dados podem circular. Assim para o sistema que foi desenvolvido optou-se por uma topologia em barramento, em que as várias luminárias (duas neste caso) possam estar “penduradas” no mesmo barramento. A arquitetura proposta é uma arquitetura *master-slave* uma vez que o protocolo que foi utilizado assim o impunha. Por uma questão de simplicidade a norma utilizada para o meio físico foi a RS-485, que para o protótipo em questão funciona perfeitamente. De referir, que o facto de haver muitos protocolos *standard* que correm sobre RS-485 também pesou para a escolha deste.

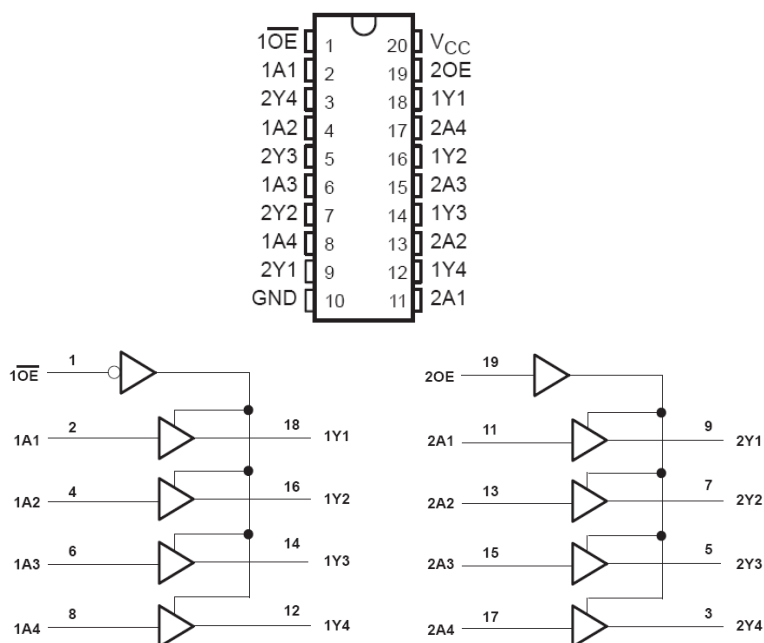
Quanto ao *master* este será constituído por um microcontrolador do mesmo tipo dos que estão incorporados nas luminárias e um circuito de seleção, que irá fazer a escolha entre comunicação via RS-485 com o barramento ou comunicação via RS-232 com um terminal (PC). Na figura seguinte apresenta-se um diagrama de blocos referente ao *master*.

Relativamente à figura 49, o seu princípio base já foi descrito anteriormente, apenas de referir a importância das resistências presentes. R1 e R2 são resistências de polarização da linha de comunicações, sendo resistências de *pull-up* e *pull-down* respetivamente. Quanto à resistência R3 é uma resistência de terminação da linha, com vista a minimizar reflexões que possam existir. Na outra terminação da linha (último *slave* “pendurado” no barramento) terá que existir uma resistência de igual valor como se irá verificar. Para finalizar, o sentido da comunicação é controlado em *software* através de um *bit* de controlo

presente no “porto c” do microcontrolador. Seguidamente irá ser feita uma breve abordagem ao integrado SN74F241 presente na figura. Na figura 50 [37] apresenta-se então a constituição do integrado em questão de maneira a perceber o princípio de funcionamento do *master*.



**Figura 49 - Diagrama de blocos referente ao master do sistema.**





**Figura 50 - Esquemático do circuito de seleção [37].**

Por observação da figura 50, facilmente se verifica que o circuito consiste em dois conjuntos de *buffers*, em que se um está ativo, o outro não está e vice-versa. Assim, facilmente se controla o sentido da comunicação manipulando um *bit* do controlador, que estará ligado aos dois pinos de *enable* do dispositivo, como se pode observar na figura 50. Assim torna-se possível realizar trocas de dados com o barramento, ou com o terminal, de uma forma independente.

Para finalizar, optou-se por esta configuração para o *master*, uma vez que torna o sistema totalmente independente de um terminal, sendo este apenas utilizado para a impressão de valores ou mensagens importantes para a manutenção e controlo do sistema, facilitando o interface sistema/utilizador. Também por uma questão de simplicidade se optou por esta configuração, uma vez que todo o *software* irá ser implementado no microcontrolador em questão, facilitando sobretudo todas as temporizações necessárias para a correta utilização do protocolo de comunicações, que seriam mais complicadas se o *software* fosse implementado diretamente no terminal, funcionando este como *master*.

Relativamente aos *slaves*, estes irão ser constituídos por duas luminárias idênticas. A estrutura da luminária é idêntica ao protótipo apresentado na figura 36. De referir que, como já foi dito anteriormente, o microcontrolador utilizado é alterado e o protótipo da figura terá dois novos sensores incorporados, que foram descritos nas secções anteriores. Na figura 51 é apresentado um esquema que representa o sistema final implementado.

Observando a figura 51 pode-se verificar que, como já foi dito, está presente uma resistência responsável por fazer a terminação da linha. Por outro lado, apesar do *standard* RS-485 utilizar apenas 2 fios, na verdade deve-se utilizar um terceiro fio (*ground*) que seja comum aos outros dois (ver capítulo 3). De notar também que o *bit* de controlo responsável pela transmissão (DE), é controlado em *software* por um *bit* presente no “porto c” do microcontrolador. Quanto à receção, o transceiver 485 está sempre pronto a receber, tornando-se num “*don’t care*” sempre que o *buffer* de transmissão seja acionado (ver tabelas 8 e 9).

Nas secções seguintes serão então apresentadas todas as funcionalidades e resultados do sistema, bem como todo o *software* necessário para o devido funcionamento do sistema.

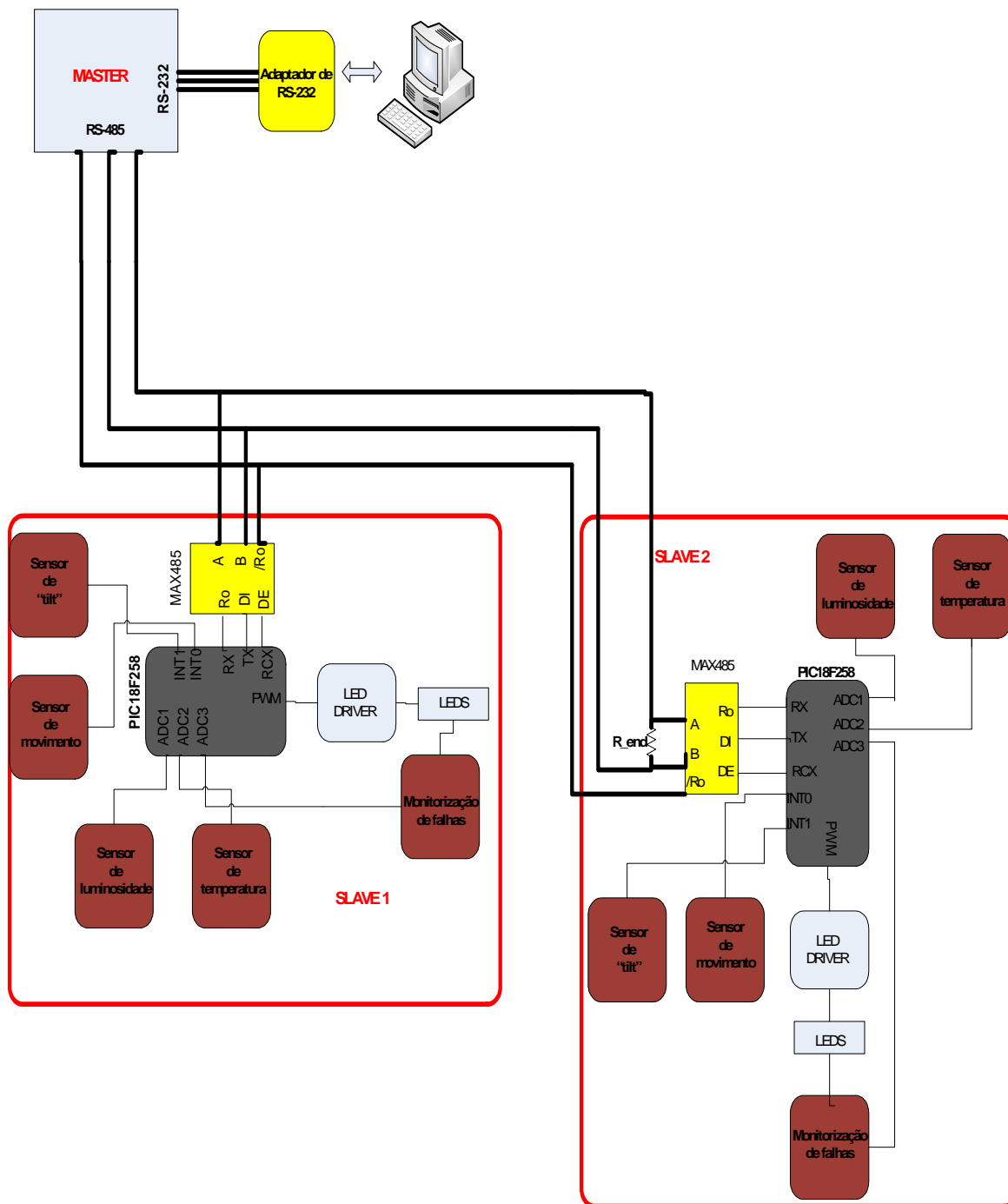


Figura 51 – Arquitetura do sistema final.

### 4.4.2 Software

Como já foi dito, nesta secção será apresentada a implementação feita a nível de *software*. No entanto, primeiro será feita uma pequena abordagem teórica ao protocolo de comunicação utilizado o Modbus.

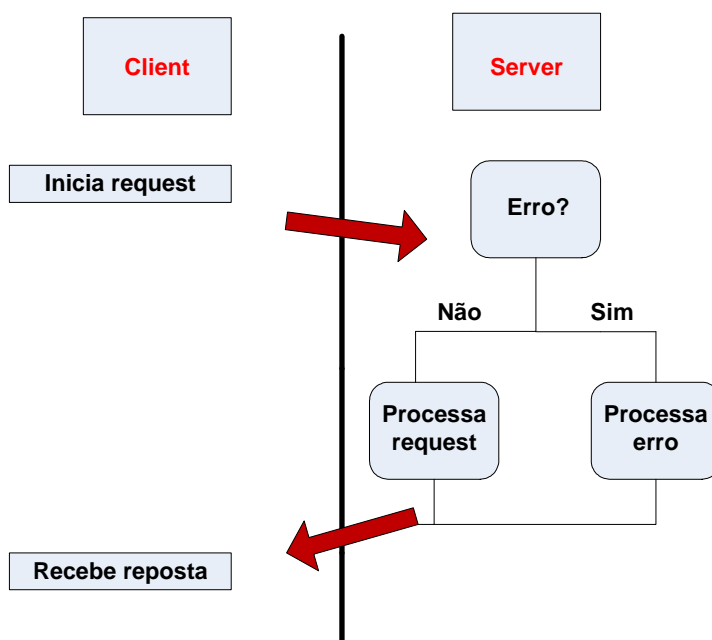
- **Modbus**

O protocolo Modbus encontra-se na camada 7, de acordo com o modelo OSI, como se pode observar na figura 52 [30]. Permite uma comunicação do tipo *master/slave* entre dispositivos ligados em diferentes tipos de barramentos (ou redes) [38].

| Layer | ISO/OSI Model |                              |
|-------|---------------|------------------------------|
| 7     | Application   | MODBUS Application Protocol  |
| 6     | Presentation  | Empty                        |
| 5     | Session       | Empty                        |
| 4     | Transport     | Empty                        |
| 3     | Network       | Empty                        |
| 2     | Data Link     | MODBUS Serial Line Protocol  |
| 1     | Physical      | EIA/TIA-485 (or EIA/TIA-232) |

Figura 52 - Localização do protocolo Modbus de acordo com o modelo OSI [30].

É um protocolo em que todas as comunicações são iniciadas pelo *master* do sistema, em que este irá construir uma trama, designada de *request* (pedido), onde estarão encapsulados todos os detalhes necessários para uma comunicação correta. Este irá ficar à espera de um *reply* (reposta) do *slave* ao qual se direccionou. Todos os outros *slaves* deverão descartar a trama. No diagrama seguinte pode-se verificar um exemplo simplificado de uma troca de mensagens entre *master* e *slave*.



### Figura 53 - Diagrama correspondente a uma comunicação Modbus.

Na figura 53 verifica-se que o protocolo requer que erros que possam acontecer durante uma comunicação sejam processados e encapsulados numa trama. Mais à frente irá ser feito um estudo acerca deste detalhe.

Atualmente o protocolo está implementado usando:

- TCP/IP sobre internet;
- Transmissão série assíncrona;
- Modbus plus, rede de alta velocidade utilizando *token passing*.

Como já foi dito, neste sistema a transmissão é série, utilizando o standard RS-485. Assim existem dois modos de transmissão Modbus em transmissão série:

- Modbus RTU (*Remote Terminal Unit*);
- Modbus ASCII (*American Code for Information Interchange*).

### • Modbus RTU

No presente sistema é utilizado o modo Modbus RTU. A trama que o protocolo Modbus especifica para a transmissão de dados denomina-se de *application data unit* (ADU). Na seguinte figura ilustram-se os vários campos da trama.

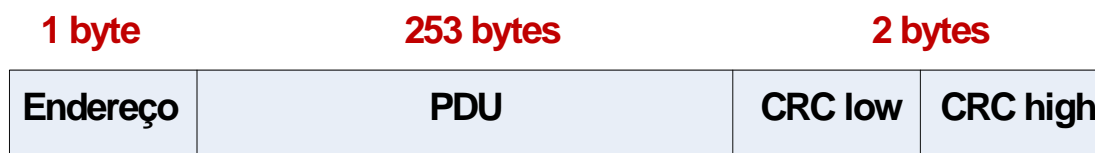


Figura 54 – *Application data unit*.

Antes de passar à especificação e descrição dos diferentes campos da ADU, é explicado como é enviado cada *byte* segundo o modo Modbus RTU e quais as temporizações necessárias que o protocolo impõe. Assim cada palavra é constituída por 11 bits, onde:

- 1 *start bit*;
- 8 *bits* de dados, em que primeiro é enviado o *bit* menos significativo e assim sucessivamente;
- 1 *bit* de paridade (1=par;0=ímpar);
- 1 *stop bit*.

O protocolo Modbus impõe que entre cada *byte* transmitido, o tempo de  $1.5 \times$  tempo de um *byte* não pode ser excedido. Entre tramas o protocolo impõe que o tempo não pode ser inferior ao tempo de  $3.5 \times$  tempo de um *byte*.

Posto isto e observando a figura 54, podemos evidenciar três campos numa trama Modbus:

- **Endereço:** Indica qual o *slave*, ao qual a trama é destinada. Possui um byte, pelo que, se pode endereçar de 1 a 255 *slaves*. De notar que o 0 corresponde a um *broadcast* (Mensagem enviada para todos os *slaves*), não sendo neste caso, enviada nenhuma resposta por parte dos *slaves*.
- **PDU (*Protocol Data Unit*):** É um campo que vai explicitar qual a ação que será efetuada (seguidamente irá ser feito um estudo mais detalhado deste campo).
- **CRC (*Cyclical Redundancy Checking*):** Verificação de erros. De referir que o CRC é calculado tendo em conta o conteúdo do PDU.

Na figura 55 ilustra-se como se divide o PDU.



**Figura 55 – Protocol Data Unit.**

Como se pode observar pela figura, o primeiro *byte* do PDU é destinado a especificar qual a função que se pretende executar. No protocolo Modbus existe um variado leque de funções que podem ser implementadas, permitindo atuar de acordo com o *request* proveniente do *master*. Na seguinte tabela apresentam-se as principais funções [38].

| Função            | Tamanho            | Tipo            |
|-------------------|--------------------|-----------------|
| Discretes inputs  | 1 bit              | Só de leitura   |
| Coils             | 1 bit              | Escrita/leitura |
| Holding Registers | Palavra de 16 bits | Escrita/leitura |
| Input Registers   | Palavra de 16 bits | Só de leitura   |

Tabela 15 – Principais funções em Modbus [38]

De referir que no sistema implementado apenas são utilizadas duas funções (*coils* e *holding registers*). A cada variável do sistema terá que ser feita uma associação em *software*, consoante seja ela uma *coil* ou uma *holding register*. De notar que no campo de dados (figura 55) existem campos que são especificados pelo protocolo consoante a função que é escolhida. Seguidamente será feito um estudo detalhado de um pedido e de uma resposta para as funções que são utilizadas.

- **Coils**

Esta função destina-se a fazer a leitura ou a escrita de um comando de on/off, ou seja, permite verificar se um certo sensor está ligado ou desligado ou permite ligar ou desligar um sensor.

- **Leitura de coils**

Na figura seguinte pode-se verificar como é encapsulada a informação num pedido para a função de leitura.

|                      |                           |                              |
|----------------------|---------------------------|------------------------------|
| <b>1 byte (0x01)</b> | <b>2 bytes</b>            | <b>2 bytes</b>               |
| <b>Função</b>        | <b>Endereço de início</b> | <b>Número de coils a ler</b> |

Figura 56 – Exemplo de um pedido de uma leitura de *coils*.

Da figura pode-se observar que o valor 1 é o especificado pelo protocolo para identificar que a ação a ser feita é a leitura de uma ou mais *coils*. O segundo campo diz respeito ao endereço de início que é o que vai fazer a associação às variáveis que se pretendem ler. De notar que estes endereços tem que ser especificados em *software* para cada uma das variáveis do sistema. Por último, de referir que este campo pode tomar

valores entre 0x0000 e 0xFFFF (em hexadecimal) [38]. O último campo diz respeito ao número de *coils* que se quer ler. Pode tomar valores entre 1 e 2000 (0x07D0) [38].

Na figura 57 verifica-se como deve ser encapsulada uma resposta Modbus para esta função.

| <b>1 byte (0x01)</b> | <b>1 bytes</b>  | <b>N bytes</b>   |
|----------------------|-----------------|------------------|
| Função               | Número de bytes | Leitura de dados |

**Figura 57 – Exemplo de uma resposta de uma leitura de *coils*.**

Na figura anterior verifica-se que no primeiro campo irá ser encapsulado o valor da função que foi pedida (a nível do pedido), no segundo campo é enviada informação acerca do número de *bytes* que vão ser lidos. De notar que uma *coil* diz respeito a 1 *bit*. Sendo assim, para fazer o cálculo do número de *bytes* terá que se dividir a quantidade de *coils* a ser lida (que vem do pedido) por 8. Se o resto da divisão for igual a zero o resultado será N *bytes*. Se o resto for diferente de 0 o resultado será N+1 *bytes* [38]. Por fim, no último campo serão encapsulados os valores lidos de cada *coil*. De referir que na construção da mensagem de resposta está implícito o processamento da mensagem de pedido (a nível do *slave*).

○ ***Escrita de uma única coil***

Na figura seguinte observa-se como é encapsulada a informação numa mensagem de pedido para a função de escrita de uma *coil*.

| <b>1 byte (0x05)</b> | <b>2 bytes</b>    | <b>2 bytes</b> |
|----------------------|-------------------|----------------|
| Função               | Endereço de saída | Valor da saída |

**Figura 58 – Exemplo de um pedido de uma escrita de uma *coil*.**

Observando a figura pode-se inferir que o valor 5 corresponde à função de escrita de uma única *coil*. O segundo campo é responsável por fazer a associação à variável que irá ser colocada a “*on*” ou a “*off*”. O último campo indica se a variável é colada a ‘1’ ou a ‘0’. Se o valor da saída for igual a 0xFF00 a variável é colocada a ‘1’, se o valor da saída for igual a 0x0000 a variável é colocada a 0 [38]. De referir que primeiro se transmite a parte mais significativa do campo de endereço de saída e seguidamente a parte menos

significativa, acontecendo o mesmo para o valor de saída. Para finalizar, a resposta a um pedido de escrita consiste num eco do pedido de escrita, ou seja, o ADU de um pedido e de uma resposta numa função de escrita é igual.

- **Holding Register**

Esta função destina-se a fazer a leitura/escrita de variáveis pertinentes do sistema, em que não é apenas um *bit* que identifica o estado de determinado sensor. É uma função útil, por exemplo, para fazer a leitura do valor do detetor de falhas dos LEDs.

- **Leitura Holding Register**

Na figura seguinte verifica-se o exemplo de um pedido numa mensagem para a função de leitura de um *Holding Register*.

|                      |                           |                           |
|----------------------|---------------------------|---------------------------|
| <b>1 byte (0x03)</b> | <b>2 bytes</b>            | <b>2 bytes</b>            |
| <b>Função</b>        | <b>Endereço de início</b> | <b>Número de registos</b> |

Figura 59 – Exemplo de um pedido de uma leitura de um *holding register*.

Verifica-se que o valor 3 identifica a função de leitura de um *Holding Register*. O campo referente ao endereço de início e ao número de registos são equivalentes aos campos que se observam na figura 56. De referir, que no caso de uma leitura de uma coil, o campo número de *coils* identifica a quantidade de *coils* a ser lida, em que, uma *coil* tem o tamanho de 1 *bit*. No presente caso, cada registo tem o tamanho de 2 bytes. Para esta função o endereço de início pode tomar valores entre 0x0000 e 0xFFFF e o número de registos pode tomar valores entre 1 e 125 (0x7D) [38]. Seguidamente apresenta-se a figura que identifica a resposta para esta função.

|                      |                        |                         |
|----------------------|------------------------|-------------------------|
| <b>1 byte (0x03)</b> | <b>1 byte</b>          | <b>N bytes</b>          |
| <b>Função</b>        | <b>Número de bytes</b> | <b>Leitura de dados</b> |

Figura 60 – Exemplo de uma resposta de uma leitura de um *Holding Register*.

Apenas de referir que, como cada registo tem o tamanho de 2 bytes, o cálculo do número de bytes, é igual a duas vezes o número de registos a ser lido, que está disponível na mensagem de pedido. Assim N (na figura 60) é igual a duas vezes o número de registos.



- **Escrita de um único Holding Register**

Na figura seguinte apresenta-se um exemplo para um pedido de escrita.

|                      |                            |                         |
|----------------------|----------------------------|-------------------------|
| <b>1 byte (0x06)</b> | <b>2 bytes</b>             | <b>2 bytes</b>          |
| <b>Função</b>        | <b>Endereço do registo</b> | <b>Valor do registo</b> |

**Figura 61 – Exemplo de um pedido de uma escrita de um único registo.**

Pode-se verificar que o valor 6 identifica a função de escrita de um único registo. Os outros campos irão identificar a variável em que se quer escrever (endereço do registo) e qual o valor a ser escrito (valor do registo). A resposta será idêntica ao pedido, como já foi dito.

- **Processamento de erros**

Como já foi mostrado na figura 53, o protocolo Modbus requer que também mensagens de erro sejam processadas e enviadas como resposta, para que o *master* tenha conhecimento do que possa estar a falhar. De notar que existem dois tipos de erro: i) o primeiro diz respeito a erros de transmissão, onde se faz a verificação de erros de paridade e das temporizações impostas pelo protocolo (em caso de erro a trama deve ser descartada). ii) o segundo diz respeito a um conjunto de exceções especificadas pelo protocolo Modbus.

Na figura 62 é apresentado um fluxograma que mostra como deve ser feito o processamento de um pedido (do lado do *slave*) [38] e, caso se encontrem erros, encapsulá-los numa trama e enviar a resposta com a correspondente exceção.

Na tabela 16 são apresentados todos os códigos referentes às exceções que podem ser encontradas. De referir, que por uma questão de simplicidade, apenas serão enunciadas as exceções que foram utilizadas no sistema implementado.

De referir que todas as condições que irão gerar, ou não, uma mensagem de erro serão especificadas mais à frente, quando se tiver a apresentar a implementação do protocolo para o sistema proposto na figura 51.

Na figura 63 demonstra-se como deve ser constituída a mensagem de resposta de um *slave*, caso se verifique alguma destas exceções.

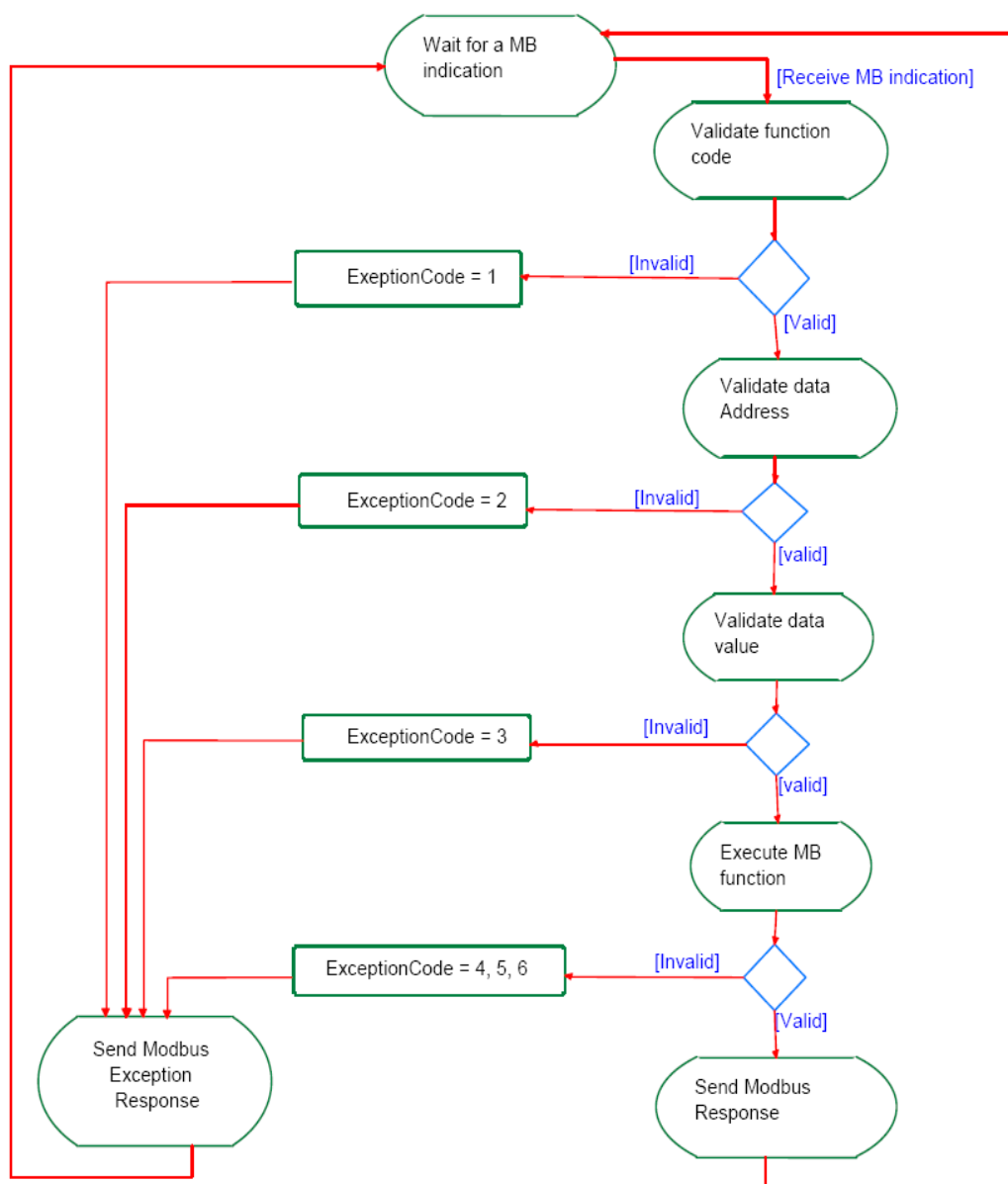


Figura 62 – Fluxograma que exemplifica o processamento de um pedido pelo *slave* [38].

1 byte

2 bytes

| Função+ 0x80 | Código de exceção |
|--------------|-------------------|
|--------------|-------------------|

Figura 63 – Mensagem de resposta do *slave* em caso de exceção.

| Código da exceção | Nome da exceção                 | Descrição da exceção  |
|-------------------|---------------------------------|---|
| 0x01              | Função não suportada            | No caso em a função não seja reconhecida pelo protocolo. Parâmetro mal atribuído.                                     |
| 0x02              | Endereço de dados mal atribuído | O endereço de dados terá que se encontrar entre um certo intervalo de valores, consoante a função que seja utilizada. |
| 0x03              | Valor de dados mal atribuído    | Consoante a função utilizada o valor dos dados terão que respeitar certas condições, caso contrário, gera um erro.    |
| 0x04              | Falha do slave                  | Caso em que se verifica falha num slave. Por exemplo, o valor lido num sensor não faz sentido.                        |

Tabela 16 – Lista de exceções do sistema

### 4.4.3 Implementação

Como já foi dito, o sistema implementado assenta na arquitetura descrita na figura 51. Nesta subsecção irá ser feita uma descrição de como foi implementado o protocolo Modbus e das funcionalidades do sistema. Anteriormente foi feita uma abordagem dos aspetos fulcrais do protocolo para que agora seja mais fácil o entendimento da sua implementação.

O sistema é constituído por duas luminárias (2 *slaves*) que, têm na sua constituição um conjunto de cinco sensores. Para o sistema proposto, o principal objetivo é estabelecer

comunicação entre o *master* e os dois *slaves* com o intuito de possibilitar a sua manutenção à distância, ou seja, fazer a leitura dos valores provenientes dos sensores. Pretende-se que as luminárias sejam suficientemente autónomas para poderem funcionar caso a comunicação falhe. Sendo assim, a maior parte do controlo terá que ser autónomo a cada luminária. Também é implementado controlo a nível do *master*. No entanto, este é um controlo simples que tem apenas por objetivo maximizar a eficiência do sistema. Por isso, o sistema terá as seguintes funcionalidades, a nível do *master*:

1. Ler valores de *coils* associadas aos sensores de *tilt* e de luminosidade;
2. Ler valores de registos associados aos sensores de temperatura e de deteção de falhas nos LEDs;
3. Permitir colocar as luminárias com fluxo de luminosidade máximo, caso se justifique;
4. Permitir que a primeira luminária que deteta luminosidade insuficiente enviar um comando (escrita de uma *coil*) para a outra luminária, para que esta ligue;
5. Permitir desligar a luminária caso se verifique um comportamento estranho pela observação dos valores dos sensores.

A nível dos *slaves* todas as suas funcionalidades já foram descritas em secções anteriores, em que, algumas delas já estavam implementadas no protótipo da figura 36. Seguidamente, enumeram-se todas essas funcionalidades:

1. Apenas ligar a luminária quando é detetado um nível mínimo de luminosidade;
2. Fazer a deteção de movimento e, quando detetado, aumentar o fluxo luminoso;
3. Fazer a leitura da temperatura dos LEDs e, como a temperatura de operação destes está situada entre  $-10^{\circ}$  e  $110^{\circ}$  [34], minimizar o fluxo luminoso caso a temperatura exceda  $110^{\circ}$ ;
4. Caso se detete inclinação da estrutura desligar imediatamente a luminária;
5. Fazer a deteção de falhas nos LEDs. Caso se detete 5 ou mais falhas ou se detetar um comportamento anormal, (tensão significativamente superior a 30V) desligar a luminária.

Uma vez identificadas todas as funcionalidades do sistema, passa-se agora à implementação do *software* que permite a comunicação entre *master* e *slave*. Assim, em primeiro lugar será feita uma abordagem ao *master* e posteriormente ao *slave*. No entanto, antes é pertinente fazer algumas observações que são comuns aos dois dispositivos.

Como já foi dito anteriormente, é necessário fazer uma associação entre as variáveis do protocolo Modbus e das variáveis do sistema, ou seja, é necessário definir um mapa de memória. Nas seguintes tabelas pode-se observar essa mesma associação, fazendo a distinção entre *coils* e *Holding Registers*.

| Coils                  |                              |
|------------------------|------------------------------|
| Variável do sistema    | Definição da variável Modbus |
| Sensor de luminosidade | Add_twilight_coil            |
| Sensor de tilt         | Add_tilt_coil                |

Tabela 17 – Associação das variáveis Modbus com as variáveis do sistema (*coils*)

| Holding Register                       |                              |
|--|------------------------------|
| Variável do sistema                    | Definição da variável Modbus |
| Detetor de falhas nos Leds             | Add_led_failure              |
| Sensor de temperatura                  | Add_temperature              |
| Variável que controla o fluxo luminoso | Add_flux_high                |

Tabela 18 – Associação das variáveis Modbus com as variáveis do sistema (*Holding Register*)

Enunciam-se agora duas especificações do sistema que são comuns tanto ao *master* como ao *slave*:

- *Baudrate* de 9600 bps;
- USART configurada para transmissão de 9 *bits*, incluindo 1 *bit* para deteção de paridade. Assim cada carater é constituído por 11 bits (9 + 1 *start bit* + 1 *stop bit*)

Em consequência do exposto anteriormente, podem agora ser definidas as temporizações impostas pelo protocolo (entre caracteres e entres tramas). Assim, o tempo que irá demorar a transmitir um carater é igual a:

$$T_{\text{carater}} = \frac{11}{\text{Baudrate}} \quad (7)$$

Sendo o baudrate de 9600 bps, tem-se que o tempo de transmissão de um carater é de 1.1ms. Assim o tempo entre cada carater não pode ser superior a 1.65ms ( $1.5 \times T_{\text{caracter}}$ ), e o tempo entre tramas não pode ser inferior a 3.9ms ( $3.5 \times T_{\text{caracter}}$ ). De referir que as temporizações serão feitas, fazendo recurso a um dos *timers* (*timer0*) disponíveis no microcontrolador PIC.

Outro aspeto que é comum ao *master* e ao *slave* são os estados que o recetor e o transmissor podem assumir. Assim para o recetor assumem-se quatro estados:

- **Recetor no estado de início**: Quando se recebe um carater neste estado inicia-se o *timer* de 3.5T carateres para identificar o final de trama. No fim da trama muda-se o estado do recetor para *idle*. De notar que neste estado a trama é descartada.
- **Recetor no estado Idle**: Quando se recebe um carater neste estado, guarda-se o carater, inicia-se o *timer* de 3.5T carateres e muda-se o estado do recetor para o estado de receção. Neste estado é identificado o início de receção de uma trama.
- **Recetor no estado de receção**: Neste estado guardam-se todos os carateres recebidos e inicia-se o *timer* de 3.5T em cada carater. De notar que entre cada carater não se pode exceder o tempo de 1.5T carateres.
- **Recetor no estado de erro**: Neste estado descarta-se toda a trama e inicia-se o *timer* de 3.5T carateres para detetar o fim de trama. No fim de trama muda-se o estado do recetor para *idle*. De notar que se entra neste estado sempre que ocorre um erro de paridade, ou sempre que a temporização de 1.5T carateres não se verifique.

Quanto ao transmissor, este pode assumir dois estados:

- **Transmissor no estado idle**: Encontra-se neste estado quando não está a transmitir.
- **Transmissor no estado de transmissão**: Neste estado envia toda a trama sequencialmente. No final inicia-se o *timer* de 3.5T carateres para garantir a temporização entre tramas.

Seguidamente serão então apresentados todos os fluxogramas (complementados com algumas explicações) pertinentes para a implementação do protocolo Modbus.

#### 4.4.4 Master

Cabe ao *master* do sistema, sempre que necessário, iniciar uma troca de informação Modbus. Nesta secção será explicado, com recurso a fluxogramas, como isso é feito. Primeiro irá ser descrita a parte inicial do protocolo, desde a sua inicialização até à preparação de um pedido. Para isso serão referidas as variáveis que são importantes para essa explicação.

Para tratar as tramas de pedido e de resposta foram definidas duas variáveis do tipo estrutura com os diferentes campos:

- Variável que identifica o *slave*;
- Array pdu onde será construído o PDU (*request*) ou onde será guardada uma trama recebida (*reply*) e com um tamanho máximo estipulado em *software*;
- Variável crc que guarda o valor de CRC;
- Variável que guarda o valor do tamanho atual do pdu;
- Variável que guarda o número de *coils*/registos a serem lidos.

De notar que todas as variáveis anteriormente enunciadas fazem parte da estrutura declarada para as variáveis *request* e *reply*. De referir que, a nível do *master*, sempre que se quiser dirigir a um *slave* diferente tem-se que inicializar o protocolo. A figura 64 mostra o fluxograma que descreve como se comporta o programa desde que se inicializa o protocolo até que uma trama de pedido seja preparada. Seguidamente irão ser enumerados alguns aspetos importantes, como auxílio ao fluxograma apresentado:

1. Aquando da iniciação do protocolo passa-se como parâmetro de uma função o identificador do *slave* ao qual a trama se deve dirigir;
2. São parâmetros de uma função o número da própria função Modbus que o protocolo deve efetuar, o endereço de início/endereço destino e endereço final/valor do registo ou *coil* (coluna da direita nas tabelas 17 e 18);

3. Segundo [38], para as funções utilizadas pelo sistema, o tamanho do PDU para a trama de pedido é sempre 5.

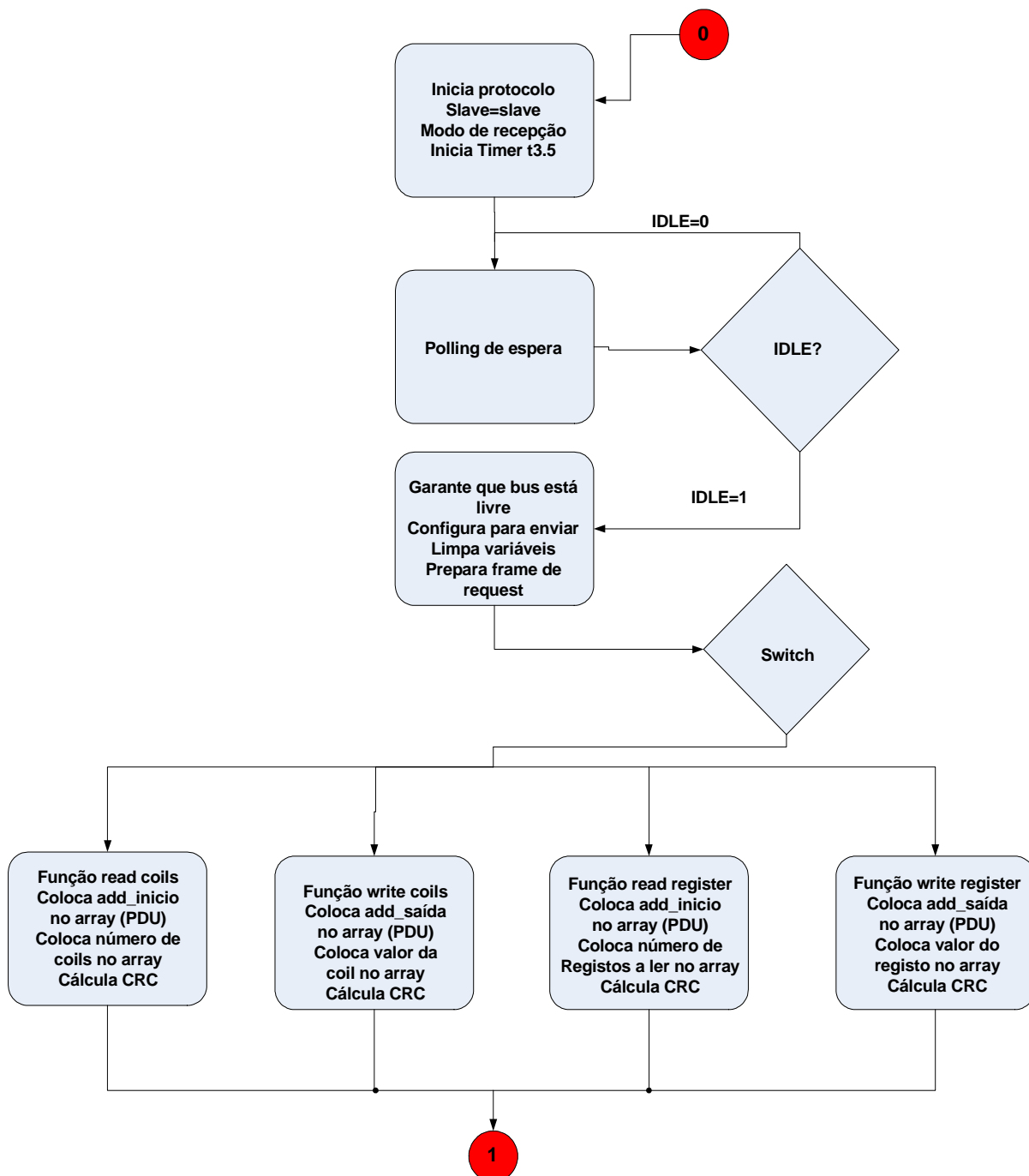
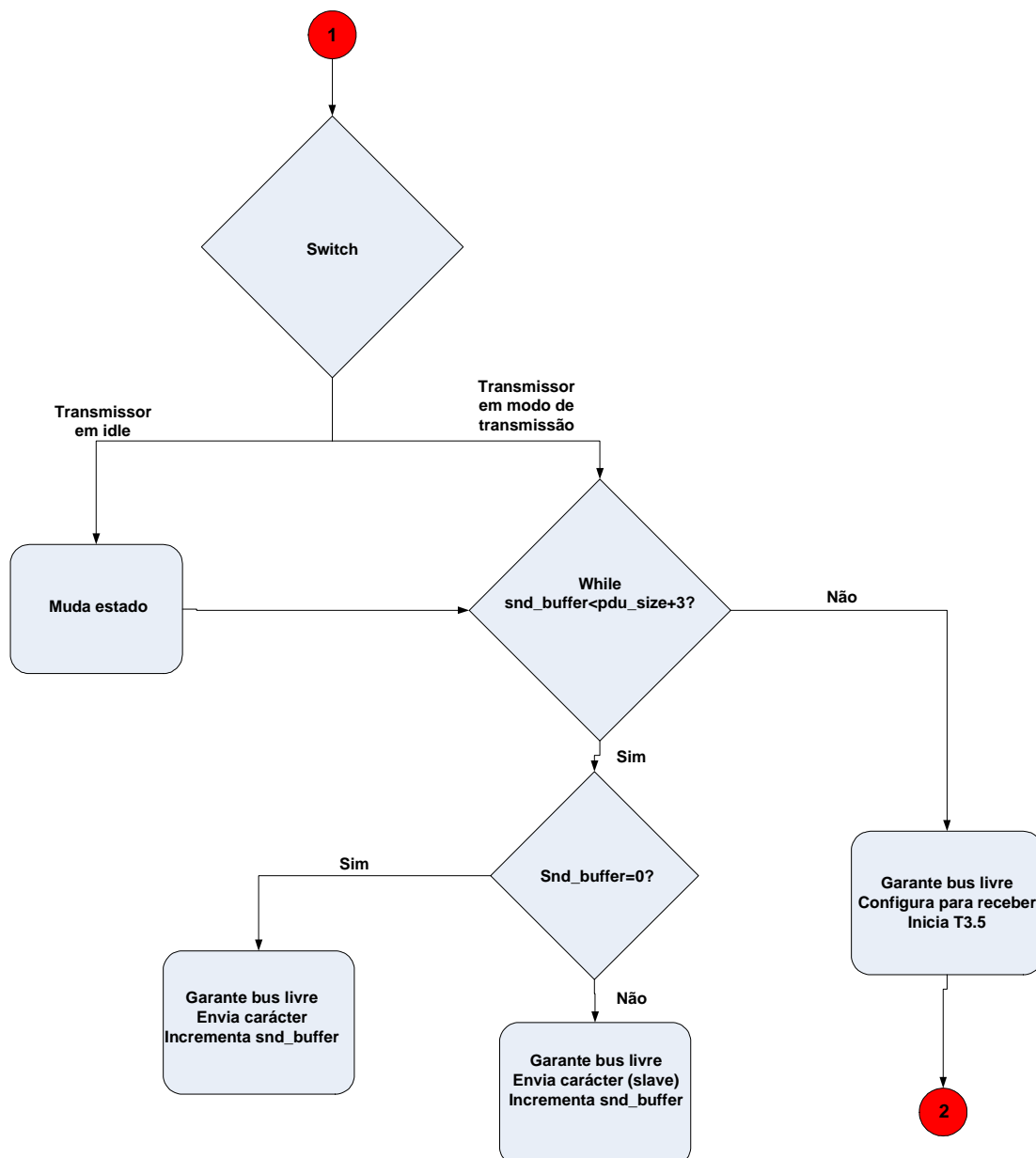


Figura 64 – Fluxograma correspondente ao início do protocolo até à preparação do pedido.



Uma vez preparada a trama de pedido, procede-se então ao seu envio. Na figura seguinte pode-se observar o fluxograma que explicita o modo como o envio de uma trama é feito. Agora enumeram-se algumas notas de auxílio à figura:

1. Deverá ser feita uma diferenciação entre o primeiro *byte* enviado e os restantes (o primeiro *byte* é a variável *slave*).
2. O valor do CRC (2 *bytes*) é também guardado no *array pdu*, como tal e visto que o primeiro carater a ser enviado é o *slave* (1 *byte*) tem que se colocar um *offset* de valor três na condição *while*, como se observa na figura 65;
3. No final inicia-se o *timer 3.5T* para garantir o tempo entre tramas.



**Figura 65 – Fluxograma relativo à transmissão de uma trama.**

Partindo do ponto 2 da figura anterior, existem alguns aspetos que têm [3]que ser explicados antes de se passar ao fluxograma da receção de uma trama. Após o envio da trama, o programa executa um ciclo de *polling* que só irá terminar quando o *timer* de  $3.5T$  (que foi inicializado no fim da transmissão) chegar ao fim. Posteriormente inicializam-se algumas variáveis pertinentes à receção e executa-se um novo ciclo de *polling*. Este ciclo é feito com o teste entre duas variáveis: uma que indica que se recebeu um carácter e como tal inicia-se o ciclo de receção; outra que indica que não houve nenhuma resposta por parte do *slave*, ou seja, indica um *timeout*. Este *timeout* é definido a partir do *timer1* do microcontrolador e tem que ser suficientemente grande para que haja tempo para haver uma resposta do *slave*. No presente sistema, o *timeout* é igual a dois segundos. Na figura 67 apresenta-se o fluxograma relativo à receção, que parte do ponto em que o ciclo de *polling* para verificar se houve ou não *timeout* chegou ao fim. Seguidamente são feitas algumas considerações para auxiliar a compreensão do fluxograma.

1. Se o primeiro carácter recebido não corresponder ao identificador do *slave* ao qual se enviou uma trama de pedido, muda-se imediatamente o estado do recetor e descarta-se a trama;
2. A leitura que é feita com o valor que se encontra no registo do *timer0*, garante que não se exceda a temporização imposta pelo protocolo, entre caracteres.

De referir que o *timer0* está configurado para, sempre que terminar a sua contagem, gerar uma interrupção. Como tal, é necessário fazer uma explicação de como é implementada a rotina de atendimento à interrupção do *timer0*. Para isso recorre-se ao fluxograma da figura 66. Da figura 66 podem-se fazer as seguintes considerações:

1. No caso de o transmissor estar no estado de transmissão, muda-se o seu estado e inicializa-se a variável. Isto acontece no final da transmissão da trama (ciclo de *polling*);
2. Se o recetor estiver no estado de erro ou de início quer dizer que a trama foi descartada. Neste caso deve-se inicializar a variável *rcv\_init* e colocar a

variável `frame_received=1` para garantir que o programa não “encrava” num ciclo de *polling* pós-receção (ver código em anexo);

3. Se o recetor estiver no estado de receção, quer dizer que foi recebida uma trama. Como tal, deve-se fazer a sua descodificação;
4. Na descodificação da trama retira-se o valor do CRC do pdu de resposta e coloca-se na variável `crc`, calcula-se novamente o CRC sobre a trama recebida e compara-se os dois valores. Se forem iguais retorna-se o valor “1”, se forem diferentes retorna-se o valor “0”;
5. No processamento da trama verifica-se se a função que vem encapsulada no pdu de resposta é suportada e, caso não seja gera-se a exceção correspondente. Caso seja uma das quatro funções suportadas, fazer a leitura das variáveis correspondentes. Se a função for de escrita apenas se imprime para o ecrã a ação que foi feita. De referir que é no processamento da trama que é feita a associação das variáveis Modbus com as variáveis do sistema (sensores), de acordo com as tabelas 17 e 18.

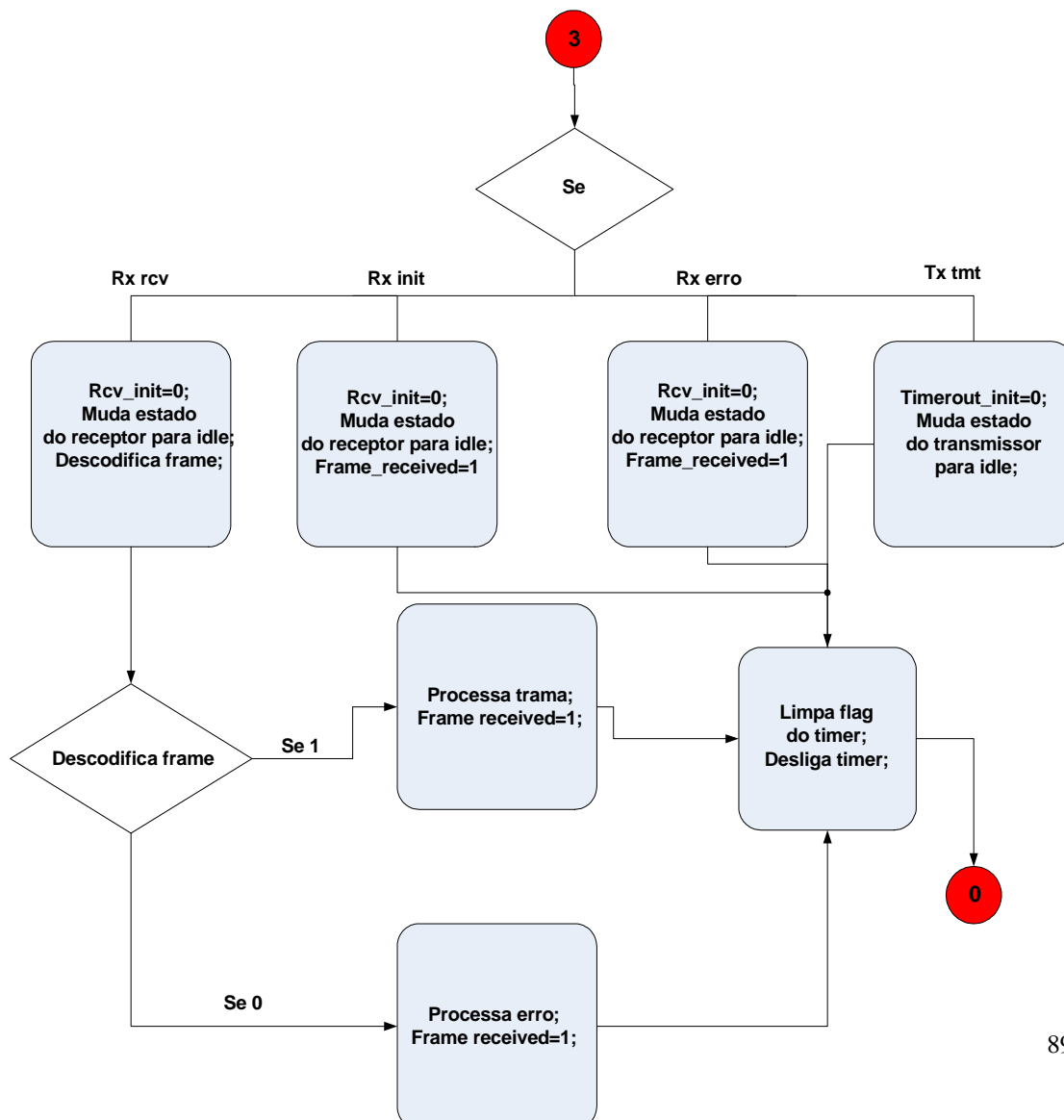
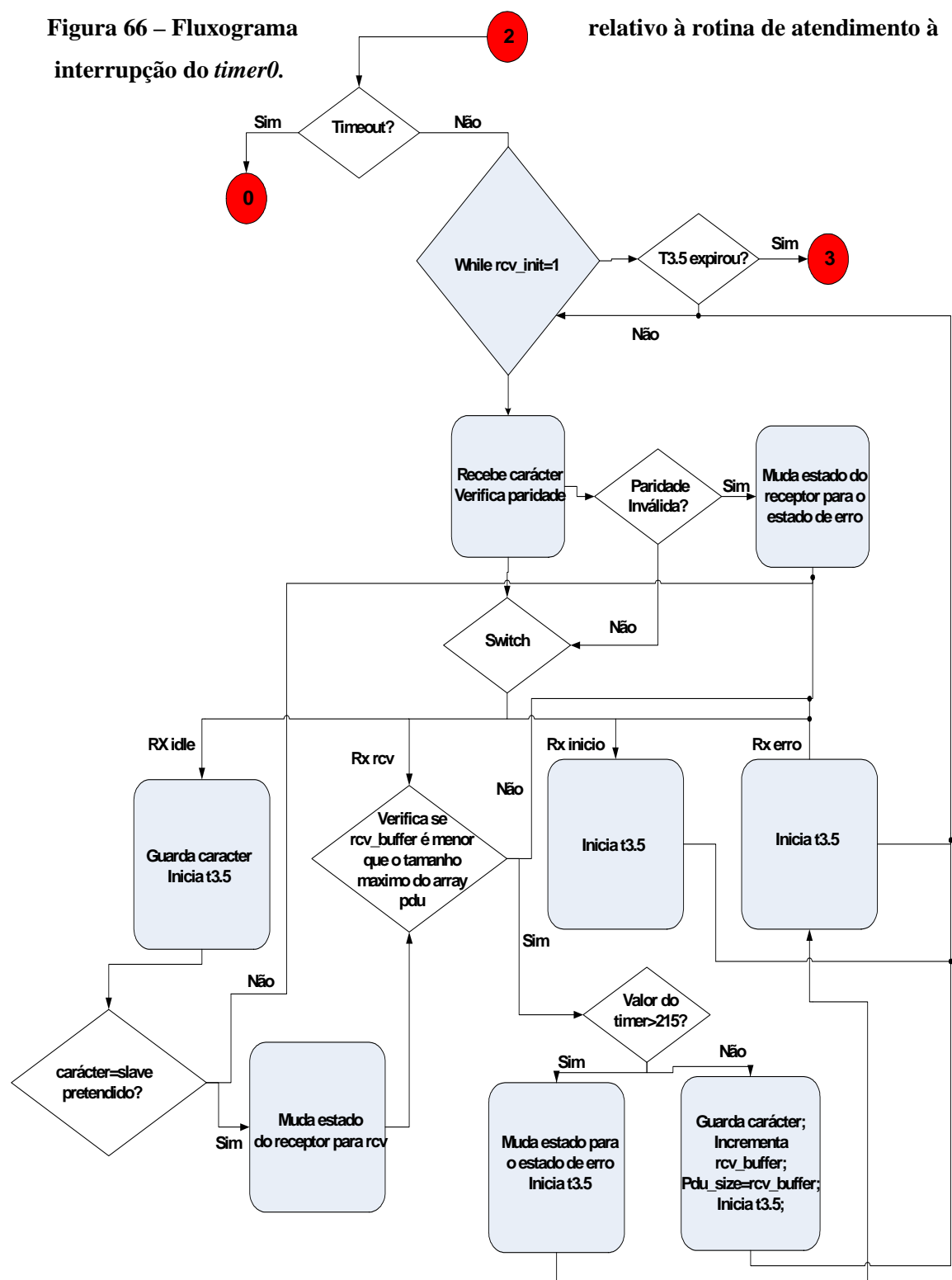


Figura 66 – Fluxograma relativo à rotina de atendimento à interrupção do *timer0*.



**Figura 67 - Fluxograma relativo à receção de uma trama.**

Seguidamente vai-se fazer uma descrição do *software* implementado do lado do *slave*. De referir que em anexos se encontra disponível o código que foi implementado, mas apenas o código relativo à implementação do protocolo.

#### **4.4.5 Slave**

Como já foi dito, no *slave* a transmissão e receção de uma trama é idêntica ao descrito nas figuras 65 e 67. No entanto, a deteção do início de receção no *slave* é diferente da do *master*, uma vez que, quer-se que o *slave* esteja a funcionar normalmente (lendo os valores dos seus sensores e atuando em conformidade) e apenas interrompa o seu funcionamento quando detetar o início de receção de uma trama. Para isso tem que se configurar o recetor do *slave* (USART) para gerar uma interrupção sempre que detetar a chegada de um carater e, sempre que for detetada uma interrupção, deve-se desligar as interrupções externas do controlador para que se trate apenas da comunicação entre *master* e *slave*. Posteriormente todo o esquema de receção é idêntico ao da figura 67, bem como o esquema da rotina de atendimento ao *timer0*. Na transmissão de dados não é necessário iniciar o *timer* de 3.5T após o envio do último carater. Sendo assim, na rotina de atendimento ao *timer0* não será necessário ter a condição do transmissor, sendo a única coisa que difere do *slave* para o *master*.

De notar que o *software* implementado para um *slave* é idêntico ao implementado para o outro, mudando apenas o seu identificador, sendo que, aquando da receção do primeiro carater e, se este não coincidir com o identificador do respetivo *slave*, a trama deve ser descartada.

No *slave*, contrariamente ao *master*, apenas se necessita de inicializar o protocolo uma vez, como é de fácil perceção. Assim, o *slave* está sempre configurado para receber, sendo apenas configurado para enviar quando tiver uma resposta disponível.

Nesta secção irá ser feito um estudo de como é processado um pedido (vindo do *master*) e, conseqüentemente é gerada uma resposta, tendo em atenção que, caso ocorram

exceções, essas exceções terão que ser encapsuladas numa trama de resposta, tal como está descrito na figura 62.

No *slave*, a descodificação de uma trama vinda do *master* é idêntica à descodificação que é feita a nível do *master*.

Seguidamente mostra-se um fluxograma que indica como é feito o processamento de uma trama com a função de leitura de *coils* encapsulada nesta. Para ajudar à compreensão deste fluxograma, enumeram-se alguns aspetos importantes:

1. O cálculo do número de *bytes* a ser lido é descrito nas páginas 76 e 78, bem como o valor máximo definido na primeira condição;
2. Pretende-se que o endereço de início que vem encapsulado na trama de pedido conjugado com o número de *coils* a ler não exceda o número total de *coils* definidas (tabela 17) e, por outro lado, o endereço de início não pode ser menor que o primeiro endereço definido no mapa de memória Modbus (que neste caso é o endereço que é associado ao sensor de *tilt*).

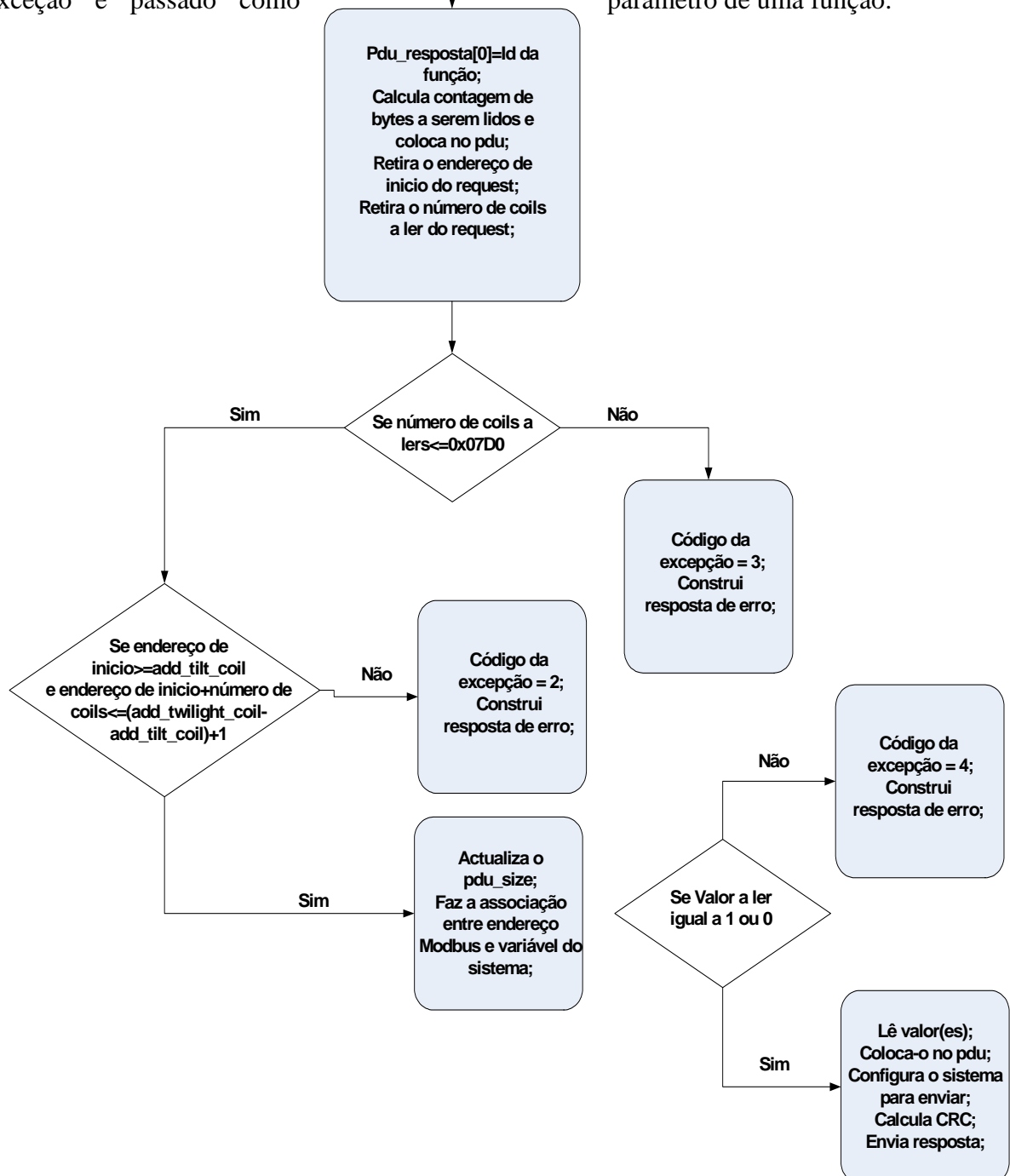
Relativamente à leitura de *holding registers* o esquema do processamento da trama é idêntico ao que é apresentado no fluxograma da figura 68. No entanto, as condições para verificar a ocorrência de uma exceção são diferentes. Enumeram-se assim as principais diferenças:

1. O cálculo do número de *bytes* e o intervalo em que o número de registos a ser lido tem que estar compreendido é descrito na página 78;
2. A segunda condição é semelhante ao caso anterior. A única diferença é que, neste caso tem que se utilizar os endereços Modbus relativos aos *holding registers* (tabela 18);
3. Caso a leitura seja referente ao sensor de temperatura, o valor lido por este tem que estar compreendido entre -10° e 110°, caso contrário, gera uma exceção de código 4 (tabela 16);
4. Se for uma leitura referente à deteção de falhas nos LEDs, o valor da leitura tem que estar compreendido entre 0 e 6.

No que diz respeito às funções de escrita o processamento é significativamente mais fácil, uma vez que, a trama de resposta é idêntica à trama de pedido. Assim apenas se tem que fazer a associação das variáveis Modbus com as variáveis dos sensores. No entanto existem algumas considerações a serem feitas:

1. A função de escrita apenas é utilizada para forçar um valor no sensor de luminosidade (forçar a ligação da luminária ou o contrário) ou para impor o aumento do fluxo luminoso;
2. No que diz respeito à escrita de uma *coil*, o valor de saída apenas pode tomar dois valores (ver página 77). Caso contrário, é gerada uma exceção. O endereço da saída (figura 58) tem que fazer parte do mapa de memória (*coils*);
3. Quanto à escrita de um registo a única diferença é que o valor de saída pode tomar qualquer valor entre 0x0000 e 0xFFFF.

Para finalizar, a construção `Read_coils` de uma trama de resposta, caso ocorra uma exceção, segue a estrutura identificada na figura 63, em que o código da exceção é passado como parâmetro de uma função.



**Figura 68 – Fluxograma relativo ao processamento de um pedido de leitura de *coils*.**

### **4.5 Em resumo**

O sistema implementado inclui melhoramentos a nível da deteção de falhas dos LEDs e a inclusão de um detetor de inclinação relativamente ao protótipo de referência. A nível de comunicação o sistema permite agora fazer a leitura de todos os sensores presentes no sistema, ou seja, a avaliação do seu comportamento pode agora ser feito à distância.

Relativamente ao detetor de falhas dos LEDs teve-se que fazer a substituição dos dispositivos responsáveis pela proteção de falhas destes (PLEDS), uma vez que a sua tensão de disparo era alta, limitando assim o número de falhas que se podiam detetar. A solução apresentada passa por fazer a leitura, em tensão, aos terminais da associação de LEDs. De notar que tem que ser introduzida alguma eletrónica para fazer a passagem dos níveis de tensão aos terminais da associação para os níveis de tensão suportados pelo PIC. De referir que a tensão aos terminais dos LEDs é bastante sensível ao aumento de temperatura, como tal, é necessária que a placa de LEDs esteja minimamente estabilizada.

No que diz respeito ao sensor de *tilt*, o sensor utilizado tem um princípio de funcionamento bastante simples. Este tem que ser calibrado com uma inclinação inicial de modo a que se garanta que não está ativo. Caso ocorra inclinação, e esta seja suficiente, o sensor irá ativar. De referir que é necessário uma configuração utilizando três sensores (o que não representa nenhum problema devido ao seu baixo custo individual) para formar o sensor final, uma vez que se quer que a deteção seja omnidirecional. Assim o funcionamento do sensor pode ser comparado ao funcionamento de três “interruptores”



dispostos em paralelo. Por conseguinte, o sensor deve ser ligado a um pino de interrupção externa do microcontrolador para que, caso ocorra inclinação, esta seja prontamente detetada. No entanto é aconselhado um estudo do comportamento do sensor em situações reais.

O sistema final é constituído por um *master* e por duas luminárias que irão funcionar como *slaves*. A arquitetura do sistema é uma arquitetura *master-slave* apresentando uma topologia em barramento. A transmissão de dados é em série, utilizando o *standard* RS-485. O *master* permite fazer a escolha entre comunicação via RS-485 e RS-232, comunicando com o barramento ou com o terminal, respetivamente. Os *slaves* apresentam na sua constituição um conjunto de cinco sensores. O protocolo de comunicações utilizado foi o Modbus, garantindo que as trocas de dados se façam de uma forma correta. A velocidade de transmissão dos dados foi definida com uma taxa de transmissão de 9600 *bits* por segundo. De notar que apenas são utilizadas quatro funções que o protocolo dispõe. Duas funções de leitura e duas funções de escrita. Assim é possível fazer a leitura de sensores que apenas podem tomar valores de “on/off” (sensor de *tilt*) ou fazer a leitura de sensores como o sensor de temperatura ou detetor de falhas. Para verificação de erros são utilizados dois métodos: um que faz uso do cálculo da paridade par em cada *byte*, permitindo fazer uma verificação *byte a byte* à medida que estes estão a ser transmitidos; o outro método consiste em fazer o cálculo do CRC sobre todo o PDU e compará-lo com o valor recebido. O sistema de comunicação proposto tem como objetivo ser uma iniciação a sistemas deste tipo, pelo que, apenas se pretende fazer a leitura das variáveis do sistema, sendo o controlo aplicado bastante simples. Para o futuro existe um grande leque de opções de controlo que podem ser implementadas e exploradas fazendo uso da comunicação entre terminal-luminárias, permitindo aumentar ainda mais a eficiência dos sistemas de iluminação pública.



# **Capítulo V**

## **5 Conclusões**

### **5.1 Análise de sensores**

Na análise de sensores serão tidas em conta as suas condições de funcionamento, o seu comportamento bem como as suas limitações. Será feita também uma abordagem de como será possível proceder à otimização destes.

#### **1) Detetor de falhas de LEDs**

A luminária implementada no sistema proposto possui uma associação com um total de dez LEDs. Assim o detetor de falhas permite fazer uma deteção até cerca de 50% do número total de LEDs, ou seja, cinco. Todas as deteções a partir das cinco falhas são desprezadas uma vez que, para cinco falhas, a luminária deve desligar automaticamente.

As leituras são realizadas em condições pouco favoráveis, uma vez que são feitas com o fluxo ao máximo, visto que, a alimentação dos LEDs é feita com corrente pulsada. Assim verifica-se que o registo da tensão aos terminais dos LEDs é fortemente afetado pela temperatura, verificando-se pequenas flutuações nessas leituras. De referir que para a situação em que não ocorrem falhas e para a situação em que existe uma falha, as leituras feitas aproximam-se bastante dos seus limites inferiores (tensão nos LEDs diminui com o aumento da temperatura), mesmo com um ambiente relativamente estabilizado (em temperatura). Assim a solução apresentada, apesar de funcional, não é otimamente robusta. Podem ser feitas algumas otimizações, como por exemplo, introduzir a variável temperatura na deteção de falhas, ou seja, fazer a deteção em função da tensão aos terminais dos LEDs e da sua temperatura. Outra otimização que poderá ser feita é fazer a deteção de falhas com o fluxo mínimo de funcionamento da luminária, sendo necessário, fazer um estudo temporal do *duty-cycle* do sinal de alimentação dos LEDs para se saber exatamente quando deve ser feita a leitura. Desta forma minimiza-se a influência da temperatura.

### **2) Sensor de inclinação**

Relativamente ao sensor de *tilt*, este está configurado para gerar uma interrupção externa sempre que seja detetada inclinação. Os resultados obtidos são satisfatórios, uma vez que demonstram que a solução apresentada é perfeitamente funcional (em ambiente controlado). De referir que a construção de um sistema de sensores tem que ser completamente simétrica para que em direções semelhantes o ângulo de ativação seja idêntico. A calibração inicial também é de extrema importância uma vez que, tem que garantir que o sensor esteja inativo no início e por forma a garantir que os ângulos de ativação não são muito elevados (ângulos de ativação de 15° podem ser consideráveis em postes de 10m). No entanto, e uma vez que o sensor faz uso de interrupções externas, poderá apresentar um comportamento não muito favorável em condições climáticas adversas, pois poderá estar sempre a fazer o atendimento da interrupção sem o poste inclinar (poste a abanar permanentemente). No entanto e ainda neste seguimento, para que o sensor detete inclinação o poste terá que fazer deslocamentos consideráveis (consoante a sua altura). Sendo assim, eventualmente irá permanecer inclinado. Em conclusão, deverá ser feito um estudo do sensor de *tilt* em condições aproximadamente reais.

## 5.2 Análise da comunicação

Apesar de o sistema de comunicações proposto não ir de encontro às premissas do projeto LITES (*Power Line Communications*), o sistema desenvolvido será implementável em sistemas reais permitindo fazer a leitura de todas as variáveis do sistema. O sistema permite fazer pequenos controlos, nomeadamente ligar uma luminária assim que a outra detetar luminosidade mínima ou aumentar o fluxo de uma luminária caso a outra tenha sido desligada por ação de um sensor (*tilt* ou falhas nos LEDs), permitindo aumentar a robustez do sistema, fazendo uso da comunicação. No entanto, existem algumas limitações que têm que ser referidas. Em primeiro lugar, o sistema não está munido com nenhum *software* (que deve ser implementado no *master*) que permita funcionar como um *bootloader*, funciona para a PIC, ou seja, um *software* que seja capaz de receber programas e enviá-los por transmissão séria para as respetivas luminárias. Isto porque, como o sistema atual não dispõe duma implementação destas, o *software* relativo ao *master* e às duas luminárias tem que ser carregado no microcontrolador de uma forma paralela ao sistema, o que não é muito robusto para um sistema real uma vez que, sempre que seja necessário fazer alguma otimização ao *software* seria necessária a deslocação ao local do sistema para carregar os programas para os controladores. Por outro lado a taxa máxima de transferência que foi alcançada foi de 9600 *bits* por segundo, o que, para sistemas atuais já não é muito rápido, uma vez que existem sistemas capazes de operar a taxas muito mais elevadas. Quanto a este aspeto não foram feitos nenhuns testes para se saber a razão exata. No entanto, e, partindo do princípio que não é devido ao *software* implementado, isto pode-se dever ao circuito que permite fazer a seleção entre transmissão RS-232 e RS-485, pois as mudanças de estado entre *standards* podem limitar a velocidade da taxa de transmissão. No que diz respeito a possíveis interferências devido a reflexões de sinal não se pode concluir nada, pois no ambiente em que os testes foram feitos a distância entre os *slaves* eram insignificantes não ocorrendo por isso, interferências. De referir que não é feito nenhum uso da comunicação para obter controlo sobre o sensor de movimento, o que poderá ser difícil para o sistema proposto uma vez que a arquitetura proposta é *master – slave*, sendo sempre o *master* o responsável pela iniciação de uma troca de dados. Assim para fazer um controlo do movimento pode ser complicado, por isso, arquiteturas como *multimaster*

poderão ser mais aconselháveis para se fazer otimizar o desempenho do sensor de movimento, uma vez que permite que as luminárias troquem dados entre si. No entanto e, tendo em conta este protótipo, podem ser estudadas e implementadas várias estratégias de controlo que possam otimizar o desempenho do sistema, atingindo maiores eficiências.

Em conclusão, o sistema implementado é uma iniciação à integração de comunicação em sistemas de iluminação pública.

### **5.3 Trabalho futuro**

- Introduzir rotinas de controlo/sistemas eficazes para um melhor controlo térmico;
- Otimizar o detetor de falhas dos LEDs, por exemplo com o descrito em 5.1, alínea 1;
- Integração do sistema numa caixa adequada a ambiente externo para possibilitar o seu teste em ambiente real;
- Testar funcionamento do sensor de *tilt* em ambiente externo;
- Desenvolver *software*, a nível do *master*, capaz de receber programas e enviá-los, por transmissão série, para os respetivos *slaves*;
- Aumentar a taxa máxima de transferência de dados;
- Estudo e desenvolvimento de rotinas de controlo robusto, para fazer uso da comunicação, de forma a melhorar o funcionamento dos sensores de movimento.
- Aumentar as opções de controlo implementadas via *master* (uso da comunicação), nomeadamente melhorar a robustez dos sensores do sistema;
- Teste do sistema de comunicação em ambientes mais próximos dos reais.

# Referências

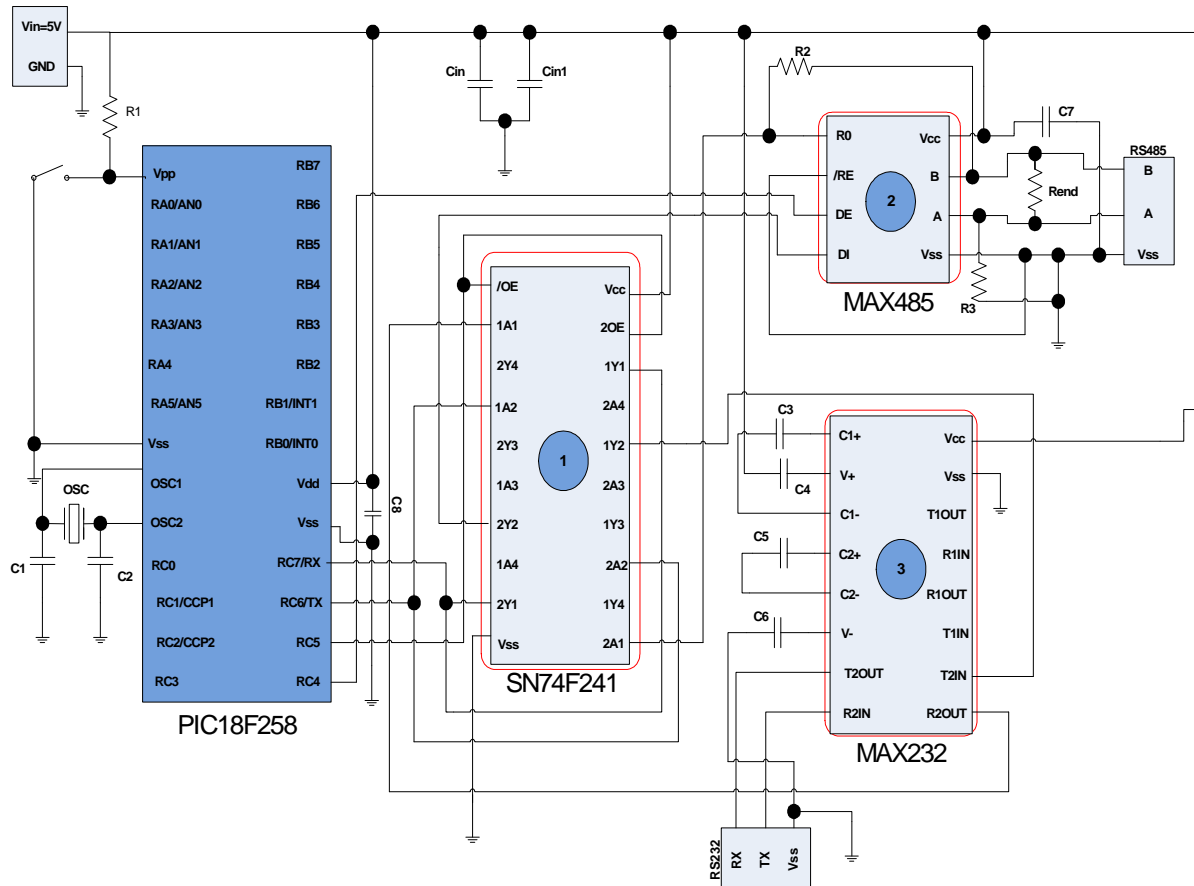
1. LITES, *Intelligent street lighting for energy saving, Presentention*.
2. J. Bullough, MS Rea, and Y. Akashi, "Several views of metal halide and high pressure sodium lighting for outdoor applications". *Lighting Research and Technology*. p. 31: 297–320. . 2009.
3. A. Amorim "Workshop Eficiência Energética na Iluminação". 2009.
4. M. Neary, M. Quijano, "SOLID STATE LIGHTING FOR INDUSTRIAL LOCATIONS." 2010.
5. LITES, *Technical Meeting*. 2010.
6. P. Boyce, "Lighting for Driving: Roads, Vehicles, Signs and Signals". 2009.
7. Sir J. Swan, "The Literary & Philosophical Society of Newcastle".
8. J. de Groot, "The High-Pressure Sodium Lamp. Deventer: Kluwer Technische Boeken BV". 1986.
9. G. Held, "Introduction to Light Emitting Diode Technology and Application". 2009.
10. *Tópicos de física no estado sólido - Apontamentos da cadeira*.
11. S. Soloman, "Sensors and Control Systems in Manufacturing".
12. J. Zhou, J. Han, W. Yan, "Experimental investigation on the performance characteristics of white LEDs used in illumination application".
13. OSRAM, "Golden Dragon Oval Plus - General Information," Application Note. 2010.
14. A. Lay-Ekuakille, F. D'Aniello, F. Miduri, D. Leonardi, A. Trotta, "Smart control of Road-based LED Fixtures for Energy Saving". 2009.
15. C. SungKwan, V. Dhingra, "Street Lighting Control based on LonWorks Power Line Communication".
16. [http://www.led-professional.com/case\\_studies/street-lighting](http://www.led-professional.com/case_studies/street-lighting).
17. O. Mannah, *Intelligent Lighting (Part 1): From Definition to Implementation*. 2010.
18. PIC18FXX8 Data Sheet.
19. L. Alves, R. Aguiar, N. Lourenço, *LITES DELIVERABLE 7.1: SENSORS TECHNOLOGIES SELECTED - A study on available technologies for efficient twilight and motion sensors*. 2010.
20. J. FRADEN, *HANDBOOK OF MODEM SENSORES - Physics, Designs, and Applications*.
21. L. Ribeiro, *Sistemas de gestão da iluminação pública*. 2009.
22. S. Inc, *Networking complete. Third edition*. 2002: San Francisco.
23. J. Alonso, L. Alonso, "A NOVEL MAC PROTOCOL FOR DYNAMIC AD HOC WIRELESS NETWORKS WITH DYNAMIC SELF-CONFIGURABLE MASTER-SLAVE ARCHITECTURE".
24. L. WuQing, "The Development of Road Lighting Intelligent Control System Based on Wireless Network Control". 2009.
25. X. Carcelle, "Power Line Communications in Practice". 2006.
26. L. Chushan, J. Wu, X. He, "Realization of a General LED Lighting System Based on a Novel Power Line Communication Technology", College of Electrical Engineering.
27. T. Instruments, *Application Report - "RS-422 and RS-485 Standards Overview and System Configurations"*. 2010.
28. D. Yulin, C. Wenchong, "The Study on Distributed Control System based on PLC". 2010.
29. MAXIM, "Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers", MAX485 datasheet.
30. "MODBUS over Serial Line - Specification and Implementation Guide, V1.02". 2006.
31. V. Souza, "Comunicação RS232 e RS485".
32. D. Lourenço, "Sistemas de Iluminação Pública com Gestão Inteligente de Consumo". 2010.
33. Littlefuse, "PLED Open LED Protectors. PLED5HT SOT 89 Series", PLED datasheet 2010.
34. OSRAM, "LW W5SM - Golden DRAGON with Chip Level Conversion (CLC)". 2009.
35. A. DEVICES, "Single-Supply, Rail-to-Rail, Low Cost, Instrumentation Amplifier", AD623 datasheet.
36. C. International, "Product Data Sheet", cw1300-1 datasheet. 2008.
37. T. Instruments, "SN54F241, SN74F241 - OCTAL BUFFERS/DRIVERS WITH 3-STATE OUTPUTS", datasheet. 1993.
38. MODBUS APPLICATION PROTOCOL SPECIFICATION. 2006.





# Anexos

- Esquemático do circuito relativo ao master – Protótipo final



1 Chip Select – Entre comunicação via rs-485 ou rs-232

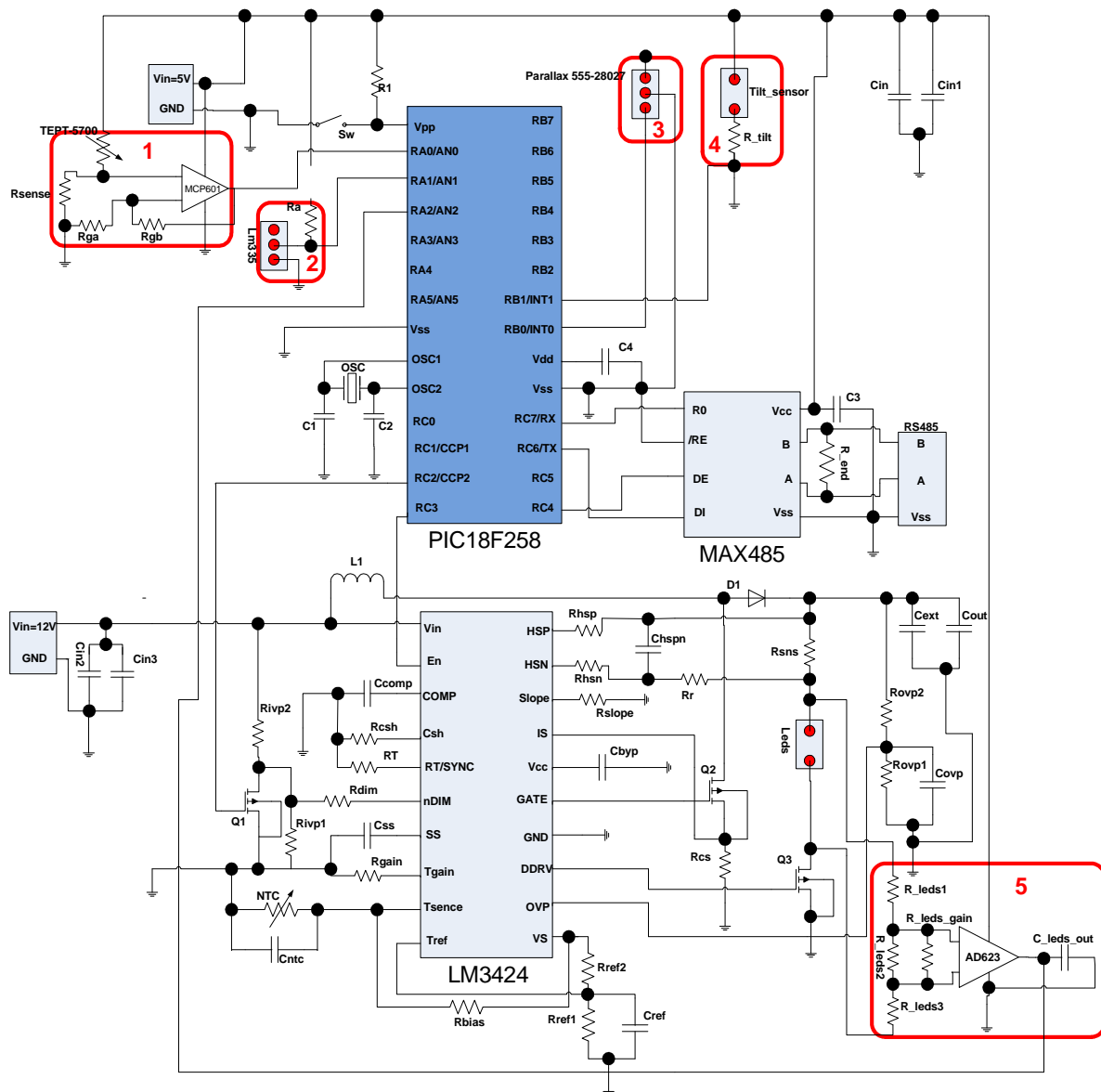
2 RS - 485 transceiver

3 Rs – 232 transceiver

C1=27p  
C2=27p  
C3=10u  
C4=10u  
C5=10u  
C6=10u  
C7=100n

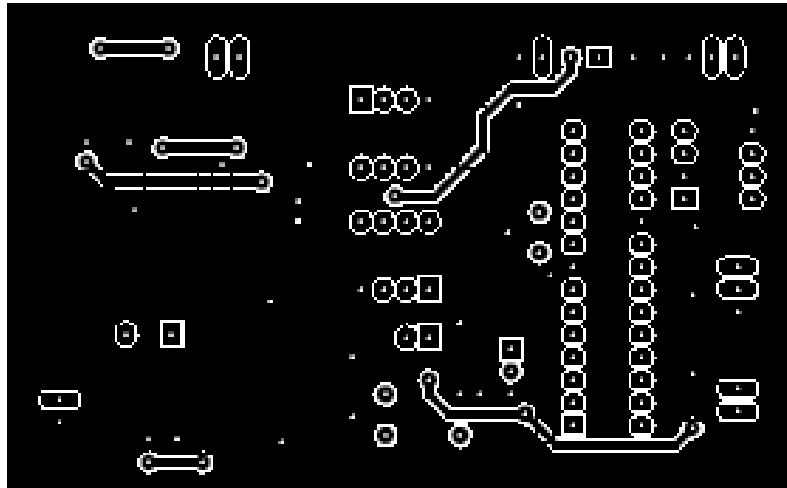
C8=100n  
Cin=100u  
Cin1=330n  
R1=390  
R2=650  
R3=650  
Rend=120

- **Esquemático do circuito relativo ao slave – Protótipo final**

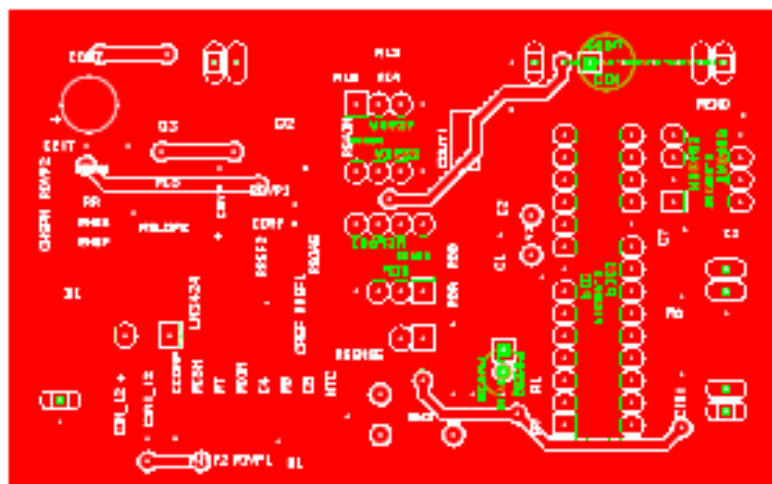


- |  |               |   |
|--|---------------|---|
| <b>1- Sensor de luminosidade</b>               | R1 = 390      | Css = 470n                                    |
| Foto transistor = TEPT- 5700                   | Cin = 100u    | Rgain = 6k34                                  |
| Rsense = 47k                                   | Cin1 = 330n   | Cntc = 100                                    |
| Rga = 10k                                      | C1, C2 = 27p  | NTC = NCP18WB473J03RB – Thermistor, SMD, 47k  |
| Rgb = 100k                                     | C4, C3 = 100n | Rhsp = 1k                                     |
|  | R_end = 120   | Rhsn = 1k                                     |
| <b>2- Sensor de temperatura</b>                | OSC = 20Mhz   | Rslope = 6k34                                 |
| Ra = 1k  | Cin2 = 100n   | Cbyp = 2.2u                                   |
| <b>3- Sensor de movimento</b>                  | Cin3 = 4.7u   | Rref1, Rref2 = 50k                            |
| <b>4- Sensor de "tilt"</b>                     | L1= 82u       | Q1 = SN7002WL6433 – MOSFET, N, 60V, SOT - 323 |
| <b>5- Sensor de detecção d falhas nos leds</b> | Rivp2 = 10k   | Q2, Q3 = STD15NF10 – MOSFET, N, D-PAK         |
| R_leds1 = 680k                                 | Rivp1 = 2k2   | D1 = C3D02060E – 600V 2A TO - 252             |
| R_leds2 = 47k                                  | Ccomp = 470n  | Rr = 10                                       |
| R_leds3 = 10k                                  | Rchs = 12k4   | Rcs = 0.16                                    |
| R_leds_gain = 100k                             | RT = 12k7     | Rbias = 16k5                                  |
| C_leds_out = 10u                               | Rdim = 10k5   | Rsns = 0.27                                   |
|  | Rovp1 = 9k1   | Rovp2 = 511k                                  |
|  | Cref = 220n   | Covp = 9k1                                    |
|  | Chspn = 100n  | Cext = 100n                                   |
|  |               | Cout = 10u                                    |

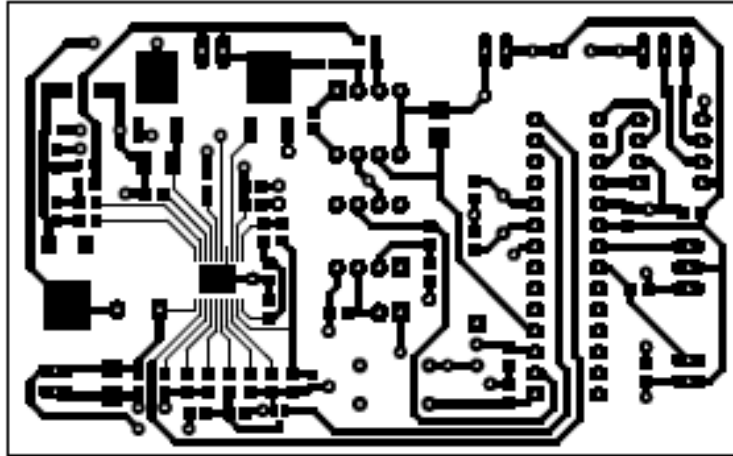
- Camada inferior da placa de circuito impresso – Protótipo final (*Slave*)



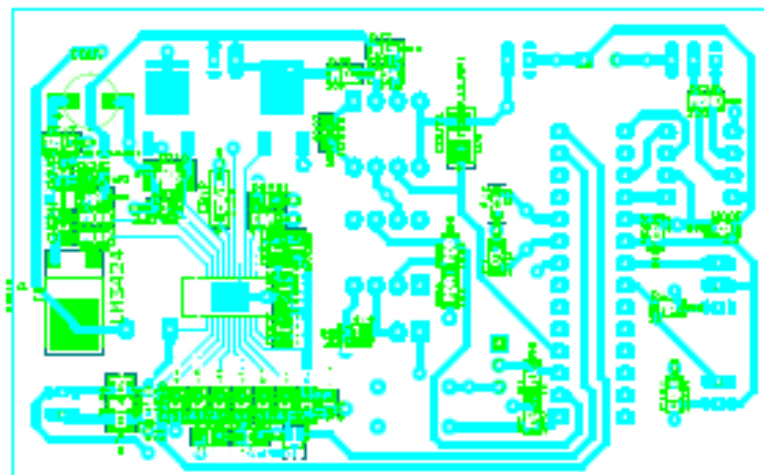
- Esquema de montagem camada inferior – Protótipo final (*Slave*)



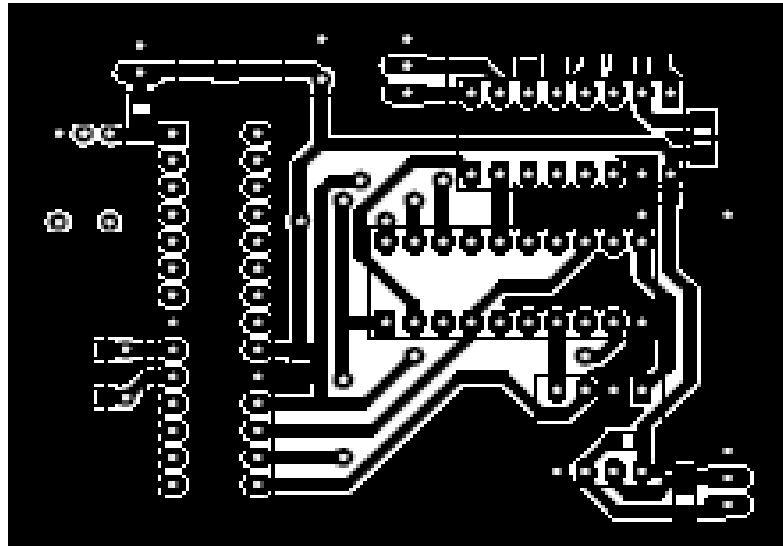
- Camada superior da placa de circuito impresso – Protótipo final (*Slave*)



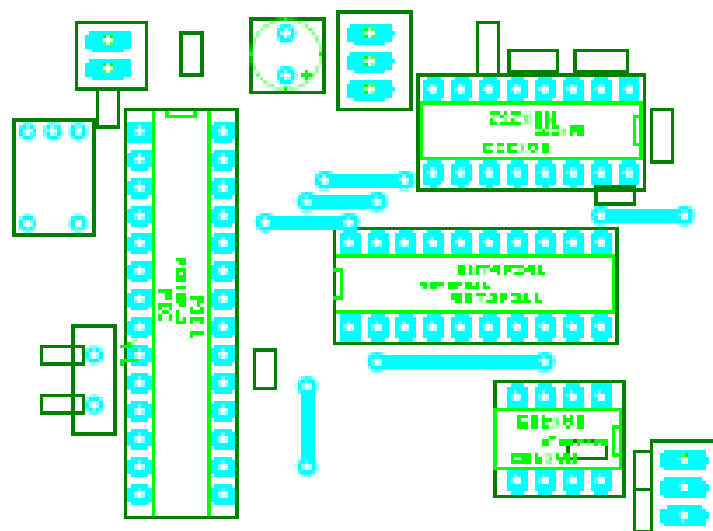
- Esquema de montagem camada superior – Protótipo final (*Slave*)



- Camada inferior da placa de circuito impresso – Protótipo final (*Master*)



- Esquema de montagem camada superior – Protótipo final (*Master*)



- **Código desenvolvido para o Master – Protótipo final**

**\* Apenas é demonstrado o código pertinente para a implementação do protocolo Modbus**

```
/*Funções de iniciação do protocolo */
eMBCode eMBInit( eMBMode eMode, UCHAR ucSlaveAddress,
                UINT ulBaudRate, eMBParity eParity
)
{

    eMBCode status= MB_ENOERR;
    if(ucSlaveAddress==MB_ADDRESS_BROADCAST||ucSlaveAddress<
        MB_ADDRESS_MIN||ucSlaveAddress>MB_AD
        DRESS_MAX)

        {
            status=MB_EX_ILLEGAL_DATA_ADDRESS;
        }

    request.slave=ucSlaveAddress;
    return status;
}
void eMBStart(void)
{

    #if MB_RTU_ENABLED>0
        eMBRTUStart();
    #endif

}

void eMBRTUStart(void)
{
    eRcvState =STATE_RX_INIT;           //No inicio está pronto a
                                         receber
    config_to_receive(); //Disable do transmissor
    RC5=1;                       //Comunicação RS-485
    idle=0;
    timer0_init(1);              //Inicia timer de t3.5
}
void MBsend(USHORT fc, SHORT bg,SHORT end)
{
    while(!idle)
        continue;
    eSndState=STATE_TX_XMIT;
    MBtimeout=0;
    rcv_init=0;
    timeout_init=0;
    USHORT snd_buffer=0;
```

```

USHORT rcv_buffer=0;
bus_free();
config_to_send(); //enable do transmissor
request.pdu_size=5;
/*-----Preparar frame de request--
-----*/
switch(fc)
{
    case MB_FUNC_READ_COILS:
        #if MB_FUNC_READ_COILS_ENABLED>0
            request.pdu[0]=MB_FUNC_READ_COILS;
            request.number=(end-bg)+1;
            request.pdu[1]=(UCHAR)(bg>>8);
            request.pdu[2]=(UCHAR)bg;
            request.pdu[3]=(UCHAR)(request.number>>8);
            request.pdu[4]=(UCHAR)request.number;
            request.crc=usMBCRC16
                (request.pdu,request.pdu_size);
            request.pdu[5]=(UCHAR)request.crc;
            request.pdu[6]=(UCHAR)(request.crc>>8);
        #endif
        break;
    case MB_FUNC_READ_HOLDING_REGISTER:
        #if MB_FUNC_READ_HOLDING_ENABLED>0
            request.pdu[0]=MB_FUNC_READ_HOLDING_REGISTER;
            request.number=(end-bg)+1;
            request.pdu[1]=(UCHAR)(bg>>8);
            request.pdu[2]=(UCHAR)bg;
            request.pdu[3]=(UCHAR)(request.number>>8);
            request.pdu[4]=(UCHAR)request.number;
            request.crc=usMBCRC16
                (request.pdu,request.pdu_size);
            request.pdu[5]=(UCHAR)request.crc;
            request.pdu[6]=(UCHAR)(request.crc>>8);
        #endif
        break;
    case MB_FUNC_WRITE_SINGLE_COIL:
        #if MB_FUNC_WRITE_COIL_ENABLED >0
            request.pdu[0]=MB_FUNC_WRITE_SINGLE_COIL;
            request.pdu[1]=(UCHAR)(bg>>8);
            request.pdu[2]=(UCHAR)bg;
            if(end==1)
            {
                request.pdu[3]=0xff;
                request.pdu[4]=0x00;
            }
            else
            {
                request.pdu[3]=0x00;

```

```

        request.pdu[4]=0x00;
    }
    request.crc=usMBCRC16
        (request.pdu,request.pdu_size);
    request.pdu[5]=(UCHAR)request.crc;
    request.pdu[6]=(UCHAR)(request.crc>>8);
    #endif
break;
case MB_FUNC_WRITE_REGISTER:
    #if MB_FUNC_WRITE_HOLDING_ENABLED >0
        request.pdu[0]=MB_FUNC_WRITE_REGISTER;
        request.pdu[1]=(UCHAR)(bg>>8);
        request.pdu[2]=(UCHAR)bg;
        request.pdu[3]=(UCHAR)(end>>8);
        request.pdu[4]=(UCHAR)end;
        request.crc=usMBCRC16
            (request.pdu,request.pdu_size);
        request.pdu[5]=(UCHAR)request.crc;
        request.pdu[6]=(UCHAR)(request.crc>>8);
    #endif
break;
default:
    exception=MB_EX_ILLEGAL_FUNCTION;
break;
}

/*-----Envia o
frame-----*/
switch(eSndState)
{
    case STATE_TX_IDLE:
        config_to_receive();        // disable do transmissor,
enable do recetor
        break;
    case STATE_TX_XMIT:
        while(snd_buffer<request.pdu_size+3)
        {
            if(snd_buffer==0)
            {
                bus_free();
                putch(request.slave);
                snd_buffer++;
            }
            else
            {
                bus_free();
                putch(request.pdu[snd_buffer-1]);
                snd_buffer++;
            }
        }
    }
}

```



```

        }
    }
    bus_free();
    config_to_receive();
    timer0_init(1);
    break;
}
while(!timeout_init)
    continue;

/*-----polling de espera por
reply do slave-----*/
framereceivedtimeout=0;
TMR1ON=1;
while((!RCIF)&&(!MBtimeout))
    continue;
    framereceived=0;
TMR1ON=0;
if(MBtimeout==0)
    rcv_init=1;
/*-----Receção de frame de
reply-----*/
while(rcv_init==1)
{
    unsigned short paridade;
    UCHAR character;
    if(RCIF==1)
    {
        character=getch();
        paridade=parity(character);
        if(parity_bit!=paridade)
            eRcvState=STATE_RX_ERROR;
        switch(eRcvState)
        {
            /* Se receber um caracter no estado inicial tem que
se esperar que o frame acabe (t35)*/
            case STATE_RX_INIT:
                timer0_init(1);
                break;

            /* No estado de erro, esperar ate ao
final da frame (t35)*/
            case STATE_RX_ERROR:
                timer0_init(1);
                break;
            case STATE_RX_IDLE:
                /*se receber um caracter, guarda-lo e mudar
estado*/
                timer0_init(1);

```

```

        reply.slave=character;
        if(reply.slave==request.slave)
            eRcvState=STATE_RX_RCV;
        else
            eRcvState=STATE_RX_ERROR;

        break;
    case STATE_RX_RCV:
        if( rcv_buffer < MB_MAX_DATA+7 )
        {
            if(TMR0L>215)
            {
                eRcvState=STATE_RX_ERROR;
                timer0_init(1);
            }
            else
            {
                /*coloca    caracter    no    buffer    e
inicializa timer35*/
                timer0_init(1);
                reply.pdu[rcv_buffer]=character;
                rcv_buffer++;
                reply.pdu_size=rcv_buffer;

            }
        }
        else
        {
            eRcvState=STATE_RX_ERROR;
        }
        break;
    }
}

/*-----Espera ate frame ser recebida e devidamente
processada antes de enviar novo request-----*/
while((!framereceived)&&(!framereceivedtimeout))
    continue;
}

*Rotina de atendimento às interrupções geradas pelo timer0

void interrupt ISR(void)
{
    GIE=0;
    if(TMR0IF==1&&TMR0IE==1)
    {
        if(flagt35==1)

```

```

{
    if(eRcvState==STATE_RX_RCV)
    {
        eRcvState=STATE_RX_IDLE;
        if(decode_frame()==1)
            processa_frame(reply.pdu[0]);
        else
            processa_error_frame(exception);
        rcv_init=0;
        framereceived=1;
    }
    else if(eRcvState==STATE_RX_INIT)
    {
        eRcvState=STATE_RX_IDLE;
        framereceived=1;
        idle=1;
        rcv_init=0;
    }
    else if(eRcvState==STATE_RX_ERROR)
    {
        eRcvState=STATE_RX_IDLE;
        RC5=0;
        printstr("\n\r");
        printstr("Erro de transmissao");
        printstr("\n\r");
        RC5=1;
        rcv_init=0;
        framereceived=1;
    }
    else if(eSndState==STATE_TX_XMIT)
    {
        eSndState=STATE_TX_IDLE;
        timeout_init=1;
    }
    TMR0ON=0;
    flagt35=0
}
GIE=1
}

```

**\*Rotina responsável pela descodificação da trama recebida**

```

short decode_frame(void)
{
    unsigned short frame_ok;
    USHORT crc;
    crc=usMBCRC16(reply.pdu,(reply.pdu_size-2));
    reply.crc=((reply.pdu[reply.pdu_size1]<<8)
              |((reply.pdu[reply.pdu_size-2])));
    if((reply.crc)!=crc)

```

```

{
    frame_ok=0;
    exception= MB_EX_CRC_ERROR;
}
else
{
    frame_ok=1;
}
return frame_ok;
}

```

**\*Rotina responsável pelo processamento de uma trama recebida**

`void processa_frame(UCHAR fc)`

```

{
    RC5=0;                                     //seleciona o hiperterminal
    printstr("\n\r");
    printstr("Processing");
    printstr("\n\r");
    USHORT i=0;
    USHORT j=1;
    unsigned short coil_count=0;
    SHORT S_ADD;
    USHORT value;
    switch(fc)
    {
        case MB_FUNC_READ_COILS:
            S_ADD=((request.pdu[1]<<8)|(request.pdu[2]));
            for(i=0;i<request.number;i++)
            {
                if(i==((j*8)-1))
                {
                    set_coils(S_ADD+i,reply.pdu[j+1],
                               coil_count);
                    j++;
                    coil_count=0;
                }
                else
                {
                    set_coils(S_ADD+i,reply.pdu[j+1],
                               coil_count);
                    coil_count++;
                }
            }
            break;
        case MB_FUNC_READ_HOLDING_REGISTER:
            S_ADD=((request.pdu[1]<<8)|(request.pdu[2]));
            for(i=0;i<request.number;i++)
            {
                set_register(S_ADD+i,((reply.pdu[j*2]<<8|

```

```

        reply.pdu[(j*2)+1]));
        j++;
    }
    break;
case MB_FUNC_WRITE_SINGLE_COIL:
    S_ADD=((request.pdu[1]<<8)|(request.pdu[2]));
    set_coils(S_ADD,((reply.pdu[3]<<8)|
        (reply.pdu[4])),0);
    break;
case MB_FUNC_WRITE_REGISTER:
    S_ADD=((request.pdu[1]<<8)|(request.pdu[2]));
    set_register(S_ADD,((request.pdu[3]<<8)|
        (request.pdu[4])));
    break;
default:
    if(fc==0x81)
    {
        exception=reply.pdu[1];
        processa_error_frame(exception);
    }
    else if (fc==0x83)
    {
        exception=reply.pdu[1];
        processa_error_frame(exception);
    }
    else if(fc==0x85)
    {
        exception=reply.pdu[1];
        processa_error_frame(exception);
    }
    else if(fc==0x86)
    {
        exception=reply.pdu[1];
        processa_error_frame(exception);
    }
    else
    {
        exception=MB_EX_ILLEGAL_FUNCTION;
        processa_error_frame(exception);
    }
    break;
}
}

```

**\*Rotina responsável pela associação de valores modbus (registers) e valores do sistema**

```

void set_register(UCHAR ADD,USHORT value)
{
    RC5=0;

```

```

switch(ADD)
{
    case ADD_LED_FAILURE:
        falhas_leds=value;
        if(reply.slave==MB_SLAVE1)
        {
            printstr("Nr de leds danificados em L1:");
            printf("%d",falhas_leds);
            printstr("\n\r");
            if((falhas_leds==5)|| (falhas_leds==6))
            {
                flag_falhas_leds_1=1;
                printstr("L1 desligada devido a demasiadas
falhas em LEDs");
                printstr("\n\r");
            }
        }
        else if(reply.slave==MB_SLAVE2)
        {
            printstr("Nr de leds danificados em L2:");
            printf("%d",value);
            printstr("\n\r");
            if((falhas_leds==5)|| (falhas_leds==6))
            {
                flag_falhas_leds_2=1;
                printstr ("L2 desligada devido a demasiadas
falhas em LEDs");
                printstr("\n\r");
            }
        }
        break;
    case ADD_TEMPERAURE:
        temp=value;
        if(reply.slave==MB_SLAVE1)
        {
            printstr("Temperatura em L1:");
            printf("%d",temp);
            printstr("\n\r");
        }
        else if(reply.slave==MB_SLAVE2)
        {
            printstr("Temperatura em L2:");
            printf("%d",temp);
            printstr("\n\r");
        }
        break;
    case ADD_FLUX_HIGH:
        if(reply.slave==MB_SLAVE1)
        {

```

```

        printstr("Colocado fluxo maximo em L1");
        printstr("\n\r");
    }
    else if(reply.slave==MB_SLAVE2)
    {
        printstr("Colocado fluxo maximo em L2");
        printstr("\n\r");
    }
    break;
}
RC5=1;
}

```

**\*Rotina responsável pela associação de valores modbus (coils) e valores do sistema**

```

void setcoils(UCHAR ADD,USHORT value,unsigned short
coil_count)
{
    RC5=0;
    switch(ADD)
    {
        case ADD_TILT_COIL:
            tilt=value>>coil_count;
            tilt=tilt&0x0001;
            if(reply.slave==MB_SLAVE1)
            {
                if (tilt==1)
                {
                    flag_tilt1=1;
                    printstr("Assistencia necessaria em L1");
                    printstr("\n\r");
                }
            }
        else if(reply.slave==MB_SLAVE2)
        {
            if (tilt==1)
            {
                flag_tilt2=1;
                printstr("Assistencia necessaria em L2");
                printstr("\n\r");
            }
        }
        break;
        case ADD_TWILIGHT_COIL:
            if(reply.pdu[0]==MB_FUNC_READ_COILS)
            {
                if(reply.slave==MB_SLAVE2)
                {
                    lumi2=value>>coil_count;

```

```

lumi2=lumi2&0x0001;
if((lumi2==1)&&(flag_tilt2==0))

{
    printstr("L2 on");
    printstr("\n\r");
}
else
{
    printstr("L2 desligada");
    printstr("\n\r");
}
}
else if(reply.slave==MB_SLAVE1)
{
    lumi1=value>>coil_count;
    lumi1=lumi1&0x0001;
    if((lumi1==1)&&(flag_tilt1==0))

    {
        printstr("L1 on");
        printstr("\n\r");
    }
    else
    {
        printstr("L1 desligada");
        printstr("\n\r");
    }
}
}
else
{
    if(reply.slave==MB_SLAVE2)
    {
        if(value==0xFF00)
        {
            lumi2=1;
            printstr("L2 ligou");
            printstr("\n\r");
        }
    }
    else if(reply.slave==MB_SLAVE1)
    {
        if(value==0xFF00)
        {
            lumi1=1;
            printstr("L1 ligou");
            printstr("\n\r");
        }
    }
}
}

```



```

    }
    }
    break;
}
RC5=1;
}

```

**\*Módulo onde são definidas todas as variáveis globais do código**

```

#ifndef _vars_H
#define _vars_H

```

```

/* ----- Defines -----
-*/

```

```

#define MB_SLAVE1      0x01 /*id do slave 1*/
#define MB_SLAVE2      0x02 /*id do slave 2*/

```

```

/* MAX DATA SIZE */

```

```

#define MB_MAX_DATA      32 /*número máximo de bytes no
campo dados do pdu */

```

```

/*Mapeamento dos coils */

```

```

#define ADD_TILT_COIL      0x0000 /*endereço de inicio
para o coil referente ao sensor de tilt-INT1 */
#define ADD_TWILIGHT_COIL  0x0001 /*endereço de inicio
para o coil referente ao sensor de lumi-RC3 */

```

```

/*Mapeamento dos registos */

```

```

#define ADD_LED_FAILURE    0x0000 /*endereço de inicio
para o registe referente as falhas nos leds */
#define ADD_TEMPERAURE     0x0001 /*endereço de inicio
para o registo referenre à temperatura */
#define ADD_FLUX_HIGH      0x0002 /*endereço de inicio
para o registe referente ao aumento do fluxo */

```

```

/*-----Variáveis -----
-*/

```

```

typedef unsigned char      UCHAR;
typedef unsigned int       USHORT;
typedef unsigned long      UINT;
typedef signed int         SHORT;
/*variavel para a paridade */
unsigned short             parity_bit;
/*variáveis auxiliares para a contagem do timer0 */

```

```

USHORT t0Count;
unsigned short      flagt35=0;
unsigned short      flagt0=0;
/* variáveis importantes para a receção */
USHORT      framereceivedtimeout;
/* Flags relativas à espera de um reply */
USHORT      framereceived;
unsigned short      timeout_init;
USHORT      MBtimeout;
unsigned short      rcv_init;
unsigned short      idle;
/*variavel referentes aos sensores */
unsigned short      flag_tilt1;
unsigned short      flag_tilt2;
unsigned short      flag_falhas_leds_1;
unsigned short      flag_falhas_leds_2;
USHORT      lumi1, lumi2;
USHORT      tilt;
USHORT      falhas_leds;
USHORT      temp;
typedef struct
{
    UCHAR      slave;          /*slave
address*/
    UCHAR pdu      [MB_MAX_DATA+7];    /*array
referente ao pdu */
    USHORT      crc;          /*crc field*/
    USHORT      pdu_size;      /*tamnaho real
do pdu */
    USHORT      number;        /*number      of
registers ou coils*/
}struct_frame;
struct_frame request,reply;

/*Maquina de estados do recetor */
typedef enum
{
    STATE_RX_INIT,      /*!< Receiver is in initial state. */
    STATE_RX_IDLE,      /*!< Receiver is in idle state. */
    STATE_RX_RCV,        /*!< Frame is beeing received. */
    STATE_RX_ERROR      /*!< If the frame is invalid. */
} eMBRcvState;
eMBRcvState eRcvState;
/* Máquina de estados do transmissor */
typedef enum
{
    STATE_TX_IDLE,      /*!< Transmitter is in idle state. */
    STATE_TX_XMIT      /*!< Transmitter is in transfer state. */
} eMBSndState;

```

```
eMBSndState eSndState;

#endif
```

- **Código desenvolvido para o Slave - Protótipo final**

\* Apenas é demonstrado o código pertinente para a implementação do protocolo Modbus e que difere significativamente do código do master

**\*Rotina responsável pelo processamento de uma trama recebida e por construir o reply**

```
void processa_frame(UCHAR fc)
{
    USHORT i=0;
    USHORT j=1;
    unsigned short coil_count=0;
    SHORT S_ADD;
    USHORT value;
    SHORT B_COUNT;
    reply.pdu_size=0;
    switch(fc)
    {
        case MB_FUNC_READ_COILS:
            reply.pdu[0]=MB_FUNC_READ_COILS;
            request.number=((request.pdu[3]<<8)|(request.pdu[4]));
            B_COUNT=request.number/8;
            if((request.number%8)==0)
                B_COUNT=B_COUNT;
            else
                B_COUNT=B_COUNT+1;
            reply.pdu[1]=B_COUNT;
            reply.pdu_size=2+B_COUNT;
            S_ADD=((request.pdu[1]<<8)|(request.pdu[2]));
            for(i=2;i<2+B_COUNT;i++)
                reply.pdu[i]=0;
            if((0x0001<=request.number)&&
                (request.number<=MB_PDU_FUNC_READ_COILCNT_MAX))
            {
                if((S_ADD>=ADD_TILT_COIL)&&
                    (S_ADD+request.number<=(ADD_TWILIGHT_COIL-
                    ADD_TILT_COIL)+1))
                {
                    for(i=0;i<request.number;i++)
                    {
                        if(i==(j*8)-1)
                        {

```

```

        get_reply_coils(S_ADD+i,j+1,coil_count);

        j++;
        coil_count=0;
    }
    else{
        get_reply_coils(S_ADD+i,j+1,coil_count);
        coil_count++;
    }
}
}
else
{
    exception=MB_EX_ILLEGAL_DATA_ADDRESS;
    processa_error_frame(exception);
}
}
else
{
    exception=MB_EX_ILLEGAL_DATA_VALUE;
    processa_error_frame(exception);
}
}
break;
case MB_FUNC_READ_HOLDING_REGISTER:
    reply.pdu[0]=MB_FUNC_READ_HOLDING_REGISTER;
    S_ADD=((request.pdu[1]<<8)|(request.pdu[2]));
    request.number=((request.pdu[3]<<8)|
                    (request.pdu[4]));
    B_COUNT=2*request.number;
    reply.pdu[1]=B_COUNT;
    reply.pdu_size=2+B_COUNT;
    if((0x0001<=request.number)&&
        (request.number<=MB_PDU_FUNC_READ_REGCNT_MAX))
    {
        if((S_ADD>=ADD_LED_FAILURE)&&
            (S_ADD+request.number<=(ADD_FLUX_HIGH-
                ADD_LED_FAILURE)+1))
        {
            for(i=0;i<request.number;i++)
            {
                get_reply_register(S_ADD+i,j*2);
                j++;
            }
        }
        else
        {
            exception=MB_EX_ILLEGAL_DATA_ADDRESS;
            processa_error_frame(exception);
        }
    }
}

```

```

    }
else
{
    exception=MB_EX_ILLEGAL_DATA_VALUE;
    processa_error_frame(exception);
}
break;
case MB_FUNC_WRITE_SINGLE_COIL:
    reply.pdu[0]=MB_FUNC_WRITE_SINGLE_COIL;
    S_ADD=((request.pdu[1]<<8)|(request.pdu[2]));
    //output address
    value=((request.pdu[3]<<8)|(request.pdu[4]));
    //output value

    reply.pdu_size=5;
    if((value==0x0000)|| (value==0xFF00))
    {
        if(S_ADD==ADD_TWILIGHT_COIL)
        {
            get_reply_coils(S_ADD,value,0);
            reply.pdu[1]=(S_ADD>>8);
            reply.pdu[2]=S_ADD;
            reply.pdu[3]=(value>>8);
            reply.pdu[4]=value;
        }
        else
        {
            exception=MB_EX_ILLEGAL_DATA_ADDRESS;
            processa_error_frame(exception);
        }
    }
else
{
    exception=MB_EX_ILLEGAL_DATA_VALUE;
    processa_error_frame(exception);
}
break;
case MB_FUNC_WRITE_REGISTER:
    reply.pdu[0]=MB_FUNC_WRITE_REGISTER;
    S_ADD=((request.pdu[1]<<8)|(request.pdu[2]));
    reply.pdu_size=5;
    value=((request.pdu[3]<<8)|(request.pdu[4]));
    if((value>=0x0000)&&(value<=0xFFFF))
    {
        if(S_ADD==ADD_FLUX_HIGH| )
        {
            get_reply_register(S_ADD,value);
            reply.pdu[1]=(S_ADD>>8);
            reply.pdu[2]=S_ADD;
            reply.pdu[3]=(value>>8);
        }
    }
}

```

```

        reply.pdu[4]=value;
    }
    else
    {
        exception=MB_EX_ILLEGAL_DATA_ADDRESS;
        processa_error_frame(exception);
    }
}
else
{
    exception=MB_EX_ILLEGAL_DATA_VALUE;
    processa_error_frame(exception);
}
break;
default:
    exception=MB_EX_ILLEGAL_FUNCTION;
    processa_error_frame(exception);
break;
}
}

```

**\*Rotina responsável pelo processamento de uma exceção Modbus**

```

void processa_error_frame(EMBException exception)
{
    reply.pdu[0]=request.pdu[0]+0x80;
    reply.pdu[1]=exception;
    reply.pdu_size=2;
}

```

**\*Rotina responsável leitura/escrita de uma coil**

```

void get_reply_coils (UCHAR ADD,USHORT value,unsigned short
coil_count)
{
    switch(ADD)
    {
        case ADD_TILT_COIL:
            if(reply.pdu[0]==MB_FUNC_READ_COILS)
            {
                if(tilt==1||tilt==0)
                {
                    flag_tilt=flag_tilt;
                    reply.pdu[value]=reply.pdu[value]/flag_tilt;
                }
            }
            else
            {
                exception= MB_EX_SLAVE_DEVICE_FAILURE;
                processa_error_frame(exception);
            }
        }
    }
}

```

```

break;
case ADD_TWILIGHT_COIL:
    if(reply.pdu[0]==MB_FUNC_READ_COILS)
    {
        if(lumi_coil==1||lumi_coil==0)
        {
            unsigned short lumi_coil_aux;
            lumi_coil_aux=lumi_coil<<coil_count;
            reply.pdu[value]=reply.pdu[value]/lumi_coil_aux;
        }
        else
        {
            exception= MB_EX_SLAVE_DEVICE_FAILURE;
            processa_error_frame(exception);
        }
    }
    else if(reply.pdu[0]==MB_FUNC_WRITE_SINGLE_COIL)
    {
        if(value==0xFF00)
        {
            lumi_coil=1;
            flag_lumi=1;
        }
        else
        {
            lumi_coil=0;
            flag_lumi=0;
        }
    }
    if((lumi_coil!=0)&&(lumi_coil!=1))
    {
        exception= MB_EX_SLAVE_DEVICE_FAILURE;
        processa_error_frame(exception);
    }
}
break;
}
}

```

**\*Rotina responsável leitura/escrita de um register**

```

void get_reply_register(UCHAR ADD,USHORT value)
{
    switch(ADD)
    {
        case ADD_LED_FAILURE:
            if(reply.pdu[0]==MB_FUNC_READ_HOLDING_REGISTER)
            {
                if(falhas_leds>=0&&falhas_leds<=6)
                {
                    reply.pdu[value]=(falhas_leds<<8);
                }
            }
        }
    }
}

```

```

        reply.pdu[value+1]=falhas_leds;
    }
    else
    {
        exception= MB_EX_SLAVE_DEVICE_FAILURE;
        processa_error_frame(exception);
    }
}
break;
case ADD_TEMPERAURE:
    if(reply.pdu[0]==MB_FUNC_READ_HOLDING_REGISTER)
    {
        if((temp>=-40)&&(temp<=110))
        {
            reply.pdu[value]=(temp<<8);
            reply.pdu[value+1]=temp;
        }
        else
        {
            exception= MB_EX_SLAVE_DEVICE_FAILURE;
            processa_error_frame(exception);
        }
    }
    break;
case ADD_FLUX_HIGH:
    if(reply.pdu[0]==MB_FUNC_WRITE_REGISTER)
    {
        max_flux=1;
        flux_max(value);
    }
    break;
}
}

```